Kubernetes-native Sicherheit

Definition und Bedeutung

Einleitung

In den letzten 5 Jahren hat Infrastruktursoftware von den zahlreichen Innovationen im Bereich cloudnative Technologien und Architekturen profitiert. Dank dieser Ansätze können Unternehmen Anwendungen skalierbarer, flexibler und dynamischer entwickeln und ausführen, als dies bisher möglich war. Container, Microservices, deklarative APIs und Software-Primitive abstrahieren die zugrunde liegenden Computing-, Storage- und Netzwerkressourcen. In Verbindung mit CD- und DevOps-Praktiken können Unternehmen verschiedener Größen und Branchen Software schneller und zuverlässiger auf den Markt bringen.

Im Zentrum dieses systemischen Wandels steht das Open Source-Container-Orchestrierungssystem Kubernetes. Kubernetes ist der De-facto-Standard für das automatisierte Deployment und Management cloudnativer Anwendungen in der Produktion.

Durch die Einführung cloudnativer Technologien entstehen neue Sicherheitsherausforderungen, aber auch Möglichkeiten zur Verbesserung bestehender Sicherheitsstrategien. Diese cloudnative Entwicklung führt zu einer neuen Bedrohungsumgebung, deren Angriffsfläche so dynamisch, schnelllebig und aktiv ist wie die Container selbst. Anwendungen lassen sich nicht mehr übersichtlich abgrenzen und werden absichtlich in Microservices aufgeteilt, sodass die Sicherheit über miteinander verbundene Komponenten, die unabhängig voneinander arbeiten, gewährleistet werden muss. Gleichzeitig können die zentralen Merkmale von Containern von Hackern ausgenutzt werden, um Angriffe schneller, in größerem Umfang und innerhalb kürzerer Zeitspannen als bisher durchzuführen.

Ähnlich wie virtuelle Maschinen und Public Cloud-Umgebungen führen Container und Kubernetes neue Infrastrukturschichten ein, für die jeweils eigene Sicherheitsmaßnahmen erforderlich sind. Als Reaktion auf diese Veränderungen verfolgen die Produkte in der Sicherheitsbranche in der Regel einen der 3 folgenden Ansätze:

1. Host- oder netzwerkbasierte Lösungen dehnen ihren Schutz auf Container aus
2. Lösungen konzentrieren sich auf den Schutz von Container-Umgebungen
3. Lösungen konzentrieren sich auf den Schutz von Containern und Kubernetes

Vergleich der verschiedenen Ansätze zur Container-Sicherheit

Wichtige Schlussfolgerung: Die Kubernetes-native Sicherheit geht über bestehende Ansätze zur Container-Sicherheit hinaus, um Kubernetes-Umgebungen zu schützen.

Die erste Kategorie von Sicherheitstools konzentriert sich auf den Schutz cloudbasierter Umgebungen mit Kontrollen für virtuelle Maschinen und Endpunkte (oder deren Netzwerke) mit Funktionen wie Firewalling. Die meisten dieser Tools sind auf einen bestimmten Use Case ausgerichtet (beispielsweise Schwachstellenmanagement, Angriffserkennung oder Webanwendungs-Firewalls) und verlieren an Bedeutung, wenn es um die Sicherheit von Containern geht, da der Schwerpunkt auf dem Schutz von Workloads und deren Daten und nicht auf IP-Adressen oder Servern liegt.

Diese Optionen bieten keine Sicherheit über den gesamten Lifecycle cloudnativer Anwendungen und sind auch nicht für die dynamische, hoch skalierbare und kurzlebige Natur von containerisierten Anwendungen konzipiert.

Die zweite Kategorie umfasst Lösungen, die Schutz über den gesamten Container Lifecycle bieten und über eine containerzentrierte Architektur verfügen. Die meisten Tools für die Container-Sicherheit fallen in diese Kategorie. Ihre Funktionen beschränken sich auf einzelne Container, die Container Runtime oder Engine (beispielsweise Docker) und die Artefakte, die zum Ausführen dieser Container verwendet werden, also die Container Images. Der Nachteil dieses Ansatzes ist, dass diese Tools keine Sicherheit auf Kubernetes-Ebene bieten.

Die dritte Kategorie geht über containerzentrierte Lösungen hinaus und bietet Sicherheit, die speziell für Kubernetes-Umgebungen konzipiert ist. Dieser Ansatz, eine Kubernetes-native Architektur, bietet Schutz für den gesamten Lifecycle cloudnativer Anwendungen und bekämpft Bedrohungsvektoren sowohl für Container als auch für Kubernetes. Red Hat® Advanced Cluster Security for Kubernetes ist wegweisend für diesen Kubernetes-spezifischen Ansatz.

Was ist Kubernetes-native Sicherheit?

Wichtige Schlussfolgerung: Eine Sicherheitslösung muss 6 entscheidende Kriterien erfüllen, um als Kubernetes-nativ zu gelten.

Kubernetes-native Sicherheit basiert auf dem Prinzip, dass Sicherheit am effektivsten implementiert wird, wenn sie auf das System abgestimmt ist, das die containerisierten Anwendungen eines Unternehmens verwaltet. Eine Sicherheitsplattform muss die folgenden Eigenschaften aufweisen, um als Kubernetes-nativ angesehen zu werden:

1. Direkte Integration mit dem Kubernetes-API-Server, um Einblick in die Workloads und die Infrastruktur von Kubernetes zu erhalten
2. Auswertung von Schwachstellen in der Kubernetes-Software
3. Sicherheitsfunktionen, einschließlich Richtlinienverwaltung, basierend auf Ressourcen innerhalb des Kubernetes-Objektmodells, einschließlich Deployments, Namespaces, Services, Pods und andere
4. Analyse deklarativer Daten von Kubernetes-spezifischen Artefakten (wie etwa Workload-Manifesten) und Konfigurationen
5. Durchsetzung integrierter Kubernetes-Sicherheitsfunktionen für mehr Automatisierung, Skalierbarkeit und Zuverlässigkeit
6. Deployment und Ausführung als Kubernetes-Anwendung, einschließlich Integration und Unterstützung für gängige Tools in cloudnativen Toolchains

Unternehmen, die ihre Kubernetes-Umgebungen sichern möchten, sollten die Sicherheitsanforderungen in wichtigen Use Cases bewerten und priorisieren, die Transparenz, Schwachstellenmanagement, Netzwerksegmentierung, Konfigurationsmanagement, Compliance, Bedrohungserkennung und Reaktion auf Sicherheitsvorfälle umfassen. Eine Kubernetes-native Sicherheitsplattform, die diese 6 Kriterien erfüllt, kann diese und andere auf Use Cases basierende Sicherheitsanforderungen umfassend erfüllen.

Die Vorteile Kubernetes-nativer Sicherheit

Wichtige Schlussfolgerung: Eine Kubernetes-native Sicherheitslösung bietet mehrere wichtige Vorteile, die andere Lösungen nicht bieten: mehr Schutz, verringerter Zeit- und Kostenaufwand sowie ein minimales operatives Risiko.

Unternehmen, die einen Kubernetes-nativen Sicherheitsansatz verfolgen, profitieren in dreifacher Hinsicht:

1. **Erhöhter Schutz:** Kubernetes-native Sicherheit beseitigt blinde Flecken auf der Angriffsfläche des Orchestrators, sodass Sicherheitsverantwortliche kritische Schwachstellen und Bedrohungsvektoren entdecken können, die ihnen sonst entgehen würden.
2. **Verringerter Zeit- und Kostenaufwand:** Ein Kubernetes-nativer Ansatz reduziert die Gesamtinvestitionen in Zeit, Aufwand und Personal, die für das Implementieren von Sicherheitslösungen erforderlich sind. Darüber hinaus optimiert die Lösung die Sicherheitsanalyse, -untersuchung und -problembehebung, indem sie zusätzlichen, für Kubernetes-Anwendungen und -Infrastrukturen spezifischen Kontext erfasst.
3. **Minimales operatives Risiko:** Eine Kubernetes-native Sicherheitslösung minimiert die Auswirkungen auf kritische Abläufe, indem sie eine Durchsetzung ermöglicht, die von der Skalierbarkeit und Resilienz des Orchestrators selbst profitiert. Dieser Ansatz vermeidet auch Konflikte und Komplexität, die zu einem operativen Risiko für kritische Anwendungen führen können.

Mehr Schutz mit Kubernetes-nativer Sicherheit

Wenn Unternehmen von der Ausführung einzelner Container zu deren Orchestrierung mit einem System wie Kubernetes wechseln, profitieren sie von den operativen Vorteilen, dass sie auf nahezu jeder Infrastruktur ausgeführt werden können, sowie von konsistenter Service Discovery, Load Balancing, automatischen Upgrades und Rollouts. Gleichzeitig vergrößert sich mit der Einführung dieser neuen Plattform jedoch die Angriffsfläche der Infrastruktur. Schwachstellen können durch Open Source-Software, ungewohnte Konfigurationen und neu entdeckte Angriffsvektoren entstehen.

Die entwicklungs- und operationsfreundlichen Abstraktionen, die Kubernetes zur Verfügung stellt, können leicht zu blinden Flecken für traditionelle Sicherheitsverantwortliche führen. Container und die Anwendungen, aus denen sie bestehen, sind nicht mehr an physische Grenzen und Ressourcen wie Server gebunden. Da Computing, Netzwerk und Storage mit Kubernetes vollständig programmatisch werden, können Sicherheitslücken ohne angemessene Transparenz bestehen bleiben.

Beseitigung blinder Flecken mit mehr Transparenz und Insights

Wichtige Schlussfolgerung: Kubernetes-native Sicherheitslösungen geben Unternehmen einen umfassenden Überblick über ihre Kubernetes-Sicherheit.

Effektive Sicherheit beginnt mit der Transparenz einer bestimmten Umgebung und ihrem allgemeinen Sicherheitszustand. Die Kubernetes-native Sicherheit bietet durch die Integration mit dem Kubernetes-API-Server direkten Einblick in Kubernetes. Diese Fähigkeit bietet eine Sicherheitsüberwachung nicht nur für die Container, die in Kubernetes-Clustern ausgeführt werden, sondern insbesondere für die Ressourcen und Objekte, wie Kubernetes sie sieht. Diese Abstraktionen – Deployments, Daemonsets, Services, Pods und andere Ressourcen – verweisen auf Controller, Netzwerkgruppierungen und Workloads. Die Einsicht in diese Daten liefert wichtigen Kontext auf Anwendungsebene.

Im Gegensatz dazu bieten containerzentrierte Sicherheitslösungen nur auf der Ebene einzelner, kurzlebiger Container und der dazugehörigen Images einen Überblick über die gesamte Anwendung. Red Hat Advanced Cluster Security ergänzt die Aktivitätsüberwachung innerhalb einzelner Container durch deploymentzentrierte Transparenz, die Nutzenden ein umfassendes, bedarfsgerechtes Bild von Kubernetes-Anwendungen und deren Sicherheit vermittelt.

Einblick in Konfigurationen

Durch den direkten Einblick in Kubernetes können Unternehmen die Sicherheitslage von Kubernetes selbst bewerten, indem sie Konfigurationen und Einstellungen evaluieren. Kubernetes-native Sicherheit automatisiert Richtlinienprüfungen, um bewährte Best Practices für die Sicherheit durchzusetzen und Fehlkonfigurationen zu erkennen, die das Risiko einer Offenlegung von Anwendungsdaten erhöhen können. Red Hat Advanced Cluster Security bewertet kontinuierlich Hunderte von Kubernetes-Konfigurationen, um schnell die allgemeine Sicherheit von Kubernetes-Umgebungen zu ermitteln. Diese Konfigurationen umfassen unter anderem:

- Rollenberechtigungen
- Zugriff auf Secrets
- Zugelassener Netzwerkverkehr
- Einstellungen für Control Plane-Komponenten
- Weitere Einstellungen

Im Vergleich dazu konzentrieren sich containerzentrierte Lösungen auf die Konfigurationen der Container Runtime Engine und einzelner Container (wie beispielsweise Container-Berechtigungen und Kernel-Funktionen). Da sie möglicherweise Kategorien von Kubernetes-Kontrollen übersehen, können unbemerkte unsichere Konfigurationen zu Exploits führen.

Einblick in Compliance

Parallel dazu ist Kubernetes-native Transparenz auch im Bereich Compliance erforderlich, einschließlich Branchenstandards wie PCI, HIPAA, SOC 2 und FedRAMP. Selbst wenn die Sicherheitslage eines Unternehmens gut ist, muss die Sicherheit von Kubernetes-Anwendungen und -Infrastrukturen angemessen berücksichtigt werden, damit Compliance gewährleistet ist. Audits und Zertifizierungen können verstärkte Kontrollmaßnahmen erfordern, die einen umfassenden Einblick in Kubernetes beinhalten.

Der Kubernetes-native Ansatz von Red Hat Advanced Cluster Security wendet spezielle Prüfungen auf Kubernetes-Cluster an, die automatisch bestehenden Compliance-Kontrollen zugeordnet werden, sodass Unternehmen unmittelbar einen Überblick über ihre Erfolgs- und Misserfolgsquoten erhalten. Beispiele hierfür sind das Einschränken des Netzwerkzugangs zwischen dem Internet und Services, die sensible Daten verarbeiten, oder das Ermitteln von Schwachstellen in der verwendeten Kubernetes-Version.

Containerzentrierte Sicherheit konzentriert sich auf Compliance-Kontrollen, die unabhängig vom Orchestrierungssystem sind und in der Regel auf der CIS-Benchmark (Center for Internet Security) für Docker basieren, wodurch die generelle Compliance einer bestimmten Umgebung unvollständig bewertet wird.

Einblick in die Isolation von Workloads

Container bieten grundsätzlich eine geringere Isolation als virtuelle Maschinen, insbesondere in Szenarien, die Mandantenfähigkeit erfordern. Einer der Hauptvorteile von Kubernetes ist die Anzahl der verfügbaren Optionen, um beim Ausführen von Containern für Isolation zwischen Workloads zu sorgen. Isolation ist ein

grundlegender Ausgangspunkt für den Schutz einer Kubernetes-Umgebung, und diese Optionen umfassen sowohl physische Grenzen (Cluster, Knoten) als auch virtuelle Grenzen. Beides kann mit Hilfe von Funktionen wie Namespaces und Netzwerkrichtlinien implementiert werden.

Kubernetes-native Sicherheit unterstützt Teams dabei, Workloads zu erkennen und zu implementieren, die innerhalb von Kubernetes mit diesen verschiedenen Optionen isoliert werden.

Containerzentrierte Sicherheit ignoriert diese unterschiedlichen Grenzen weitgehend, indem sie sich auf die Hosts konzentriert, auf denen die verschiedenen Container ausgeführt werden, ohne zu wissen, ob die Anwendungen und ihre Services ausreichend voneinander getrennt sind.

Red Hat Advanced Cluster Security bietet Insights, die:

- auf der Kenntnis von Clustern, Namespaces und Pods basieren
- Daten von den einzelnen Hosts erfassen
- sich in die Netzwerkrichtlinien von Kubernetes integrieren lassen, um ein umfassendes Verständnis dafür zu erhalten, wie Anwendungen isoliert werden

Eine effektive Sicherheitsstrategie ist nur so stark wie ihre Basis, und ein umfassender Einblick in Kubernetes ist für den erfolgreichen Schutz cloudnativer Anwendungen unerlässlich.

Entdeckung von kritischen Schwachstellen und Bedrohungsvektoren

Wichtige Schlussfolgerung: Kubernetes-native Sicherheitslösungen entdecken Kubernetes-Schwachstellen, Angriffsvektoren und Fehlkonfigurationen und schützen davor.

Ein wichtiger Aspekt bei der Sicherheit von Workloads, die auf Kubernetes ausgeführt werden, ist die Kenntnis von und der Schutz vor Bedrohungen, die für cloudnative Umgebungen spezifisch sind – Bedrohungen nicht nur für Container, sondern auch für die Kubernetes-Plattform selbst. Aufgrund des Umfangs und der Komplexität von Kubernetes entstehen zahlreiche Angriffsvektoren, die mit der Weiterentwicklung der Plattform durch die Cloud Native Computing Foundation (CNCF), die Open Source Communitys und die beitragenden Organisationen kontinuierlich zunehmen. Zu den Kubernetes-spezifischen Sicherheitsbedrohungen mit öffentlichen Auswirkungen gehören:

- Ein Kubernetes-Dashboard, das in der Cloud-Umgebung von Tesla unsicher konfiguriert war,[1] sodass Angreifer Zugang zu den Anmeldedaten von Konten erhalten und Kryptowährung erbeuten konnten.
- Ein Service bei Shopify, der anfällig für SSRF-Angriffe (Server-Side Request Forgery)[2] war, durch die ein Angreifer Kubelet-Anmeldedaten abrufen und wiedergeben könnte, um Root-Zugriff auf sämtliche Container zu erhalten.

Auch die cloudnative Community ist sich dieser besonderen Bedrohungen für Kubernetes bewusst. Bei einer ersten Sicherheitsprüfung im Jahr 2020[3] wurden Schwachstellen in Kubernetes festgestellt, einschließlich eines Bedrohungsmodells, das sich auf 8 wichtige Kubernetes-Komponenten erstreckt. Die Kubernetes-native Sicherheit in Red Hat Advanced Cluster Security baut auf diesen und anderen Forschungsarbeiten im gesamten cloudnativen Ökosystem auf, um spezielle Funktionen zum Erkennen von Schwachstellen, Fehlkonfigurationen und Exploits in Kubernetes bereitzustellen.

1 *Newman, Lily Hay: „Hack Brief: Hackers Enlisted Tesla's Public Cloud to Mine Cryptocurrency". Wired, 28. Februar 2018.*

2 *Kerner, Sean Michael: „How Shopify Avoided a Data Breach, Thanks to a Bug Bounty". eWeek, 17. Dezember 2018.*

3 *„Kubernetes Final Report". GitHub (kubernetes/community/wg-security-audit/findings), 6. August 2019.*

Erkennung Kubernetes-spezifischer Schwachstellen

Unter den vielen Komponenten von Kubernetes ist der API-Server zweifellos die kritischste, die es zu schützen gilt, da er eine zentrale Rolle beim Orchestrieren und Verwalten containerisierter Anwendungen spielt. Ein Beispiel für eine kritische Schwachstelle, die sich auf den Kubernetes-API-Server auswirkt, ist CVE-2018-1002100: Ein Angreifer könnte damit einen ganzen Cluster kompromittieren und gleichzeitig autorisierte Anfragen stellen, wodurch die Identifizierung noch schwieriger wäre. Die Schwachstelle war bis zu ihrer Entdeckung mehr als 3 Jahre nach dem ersten Release von Kubernetes in jeder Version vorhanden.

Kubernetes-native Sicherheit entdeckt automatisch bekannte Kubernetes-Schwachstellen, sodass Cluster-Administrations- und -Operations-Teams einige der wichtigsten Risiken für ihre Umgebungen bewältigen können.

Im Gegensatz dazu konzentrieren sich containerzentrierte Lösungen nur auf Schwachstellen in Container Images. Container mit weniger Schwachstellen sind zwar sicherer, aber die anfällige Kubernetes-Infrastruktur könnte sie anfällig für Kompromittierung machen. Red Hat Advanced Cluster Security identifiziert proaktiv die Cluster mit den meisten Kubernetes-Schwachstellen und ermöglicht es Nutzenden, Richtlinien auf Basis einzelner Schwachstellen festzulegen.

Erkennung Kubernetes-spezifischer Fehlkonfigurationen

Zusätzlich zu Schwachstellen können Fehlkonfigurationen zentraler Kontrollen von Kubernetes zu Exploits und zur Offenlegung sensibler Anwendungsdaten führen, insbesondere wenn die Fehlkonfiguration automatisch in Clustern propagiert werden kann. Diese Fehlkonfigurationen können entweder auf einen Benutzerfehler oder fehlende Kubernetes-Kenntnisse zurückgeführt werden. Beispiele für wichtige Fehlkonfigurationen sind die Vergabe umfassender Cluster-Administrationsberechtigungen an unqualifizierte Nutzende oder Services unter Verwendung von RBAC (Role-based Access Control) oder die unnötige Offenlegung von Kubernetes-Secrets, weil Deployments Secrets auch dann abrufen können, wenn sie nicht benötigt werden.

Red Hat Advanced Cluster Security lässt sich in diese Kontrollen integrieren, um Unternehmen schnell zu alarmieren, wenn Fehlkonfigurationen auftreten, und trägt so zu einem insgesamt verbesserten Konfigurationsmanagement bei.

Containerzentrierte Lösungen konzentrieren sich auf problematische Konfigurationen der Container Runtime oder einzelner Container und verursachen unvollständige Schutzmaßnahmen, die umgangen werden können. Selbst wenn eine Fehlkonfiguration für einen bestimmten Container behoben ist, könnte eine Fehlkonfiguration in Kubernetes es einem Angreifer ermöglichen, den Knoten zu kompromittieren, auf dem dieser Container ausgeführt wird. In dem genannten Beispiel ist der Container immer noch kompromittiert, obwohl der Container selbst korrekt konfiguriert wurde.

In Red Hat Advanced Cluster Security wird dieses Problem durch das Suchen nach Fehlkonfigurationen sowohl in Kubernetes als auch in Containern behoben.

Schutz von Ingress- und Egress-Netzwerkkommunikation

Kubernetes umfasst auch erweiterbare Netzwerkoptionen, einschließlich der Kommunikation von Pods untereinander mithilfe von Netzwerkrichtlinienspezifikationen. Diese Einstellungen haben jedoch Auswirkungen auf die Sicherheit, die leicht übersehen werden können. Wenn beispielsweise keine Kubernetes-Netzwerkrichtlinien auf einen bestimmten Pod angewendet werden, ist der gesamte Netzwerkverkehr (Ingress und Egress) erlaubt. Kubernetes wendet standardmäßig keine Netzwerkrichtlinien an.

Der Kubernetes-native Sicherheitsansatz versteht die Netzwerktopologie sofort, basierend auf der Konfiguration der Netzwerkrichtlinien, die häufig von DevOps- und Operations-Teams vorgenommen wird.

Containerzentrierte Lösungen wissen, wie der Netzwerkverkehr auf Basis von Regeln innerhalb ihrer eigenen proprietären Schnittstellen außerhalb von Kubernetes konfiguriert wird. Red Hat Advanced Cluster Security stellt sicher, dass konsistente, offene und standardisierte Regeln für den Netzwerkverkehr in Kubernetes eingehalten werden.

Der erfolgreiche Schutz containerisierter Anwendungen muss mit der Sicherheit der Container-Infrastruktur beginnen, auf der sie ausgeführt werden. Dieser Schritt erfordert die Kenntnis von kritischen Schwachstellen, Bedrohungsvektoren und Fehlkonfigurationen speziell für Kubernetes, die nur mit einer Kubernetes-spezifischen Sicherheitslösung erreicht werden kann.

Verringerung des Zeit- und Kostenaufwands mit Kubernetes-nativer Sicherheit

Einer der wichtigsten Vorteile von Kubernetes besteht darin, dass es eine einheitliche Plattform für das Provisionieren und Verwalten von Infrastruktur-Services bietet, unabhängig davon, ob sie in der Cloud oder On-Premise ausgeführt werden. So lassen sich operative Komplexität und Inkonsistenz beseitigen und die Workflows zwischen Entwicklungs-, Operations- und Sicherheitsteams optimieren, um Zeit und Kosten zu sparen. Unternehmen sollten ihre Kubernetes-Umgebungen auf ähnliche Weise sichern und Lösungen wählen, die den erforderlichen Aufwand reduzieren.

Die folgenden wichtigen Fragen sollten Sie sich bei der Bewertung von Sicherheitslösungen stellen:

- Verkürzt die Lösung den Lernprozess für Sicherheits- und DevOps-Teams?
- Kann die Lösung Teams bei der Analyse, Untersuchung und Problembehebung bei Vorfällen unterstützen?
- Bietet die Lösung Automatisierung und entsprechende Datenerfassung?

Eine Kubernetes-native Architektur erfüllt diese Anforderungen in mehrfacher Hinsicht.

Vereinfachter Lernprozess für Teams

Wichtige Schlussfolgerung: Kubernetes-native Sicherheitslösungen vereinfachen die Kubernetes-Sicherheit durch die Integration mit offenen, standardisierten Abstraktionen und Tools, die bereits von DevOps-Teams verwendet werden.

Komplexität ist der größte Feind von Sicherheit. Kubernetes enthält zahlreiche Komponenten und dazugehörige Tools. Daher müssen Teams neue Kompetenzen erwerben und neue Workflows in Entwicklung und Operations einführen. Zusätzliche Komplexität von einer Sicherheitslösung, die völlig unabhängig von diesen neuen Investitionen in Toolchains und Prozesse ist, ist hier völlig fehl am Platz.

Nutzung eines einzelnen, vertrauten Frameworks

Eine Kubernetes-native Architektur minimiert die Anzahl der neuen Schnittstellen, Konfigurationen und Ressourcenmodelle, die Nutzende für die Sicherheit ihrer cloudnativen Workloads erlernen müssen. Zum Entwickeln und Bereitstellen containerisierter Anwendungen verwenden Entwicklungs- und Operations-Teams bereits standardisierte Abstraktionen wie die Kubernetes-Manifeste. Die Sicherheitsfunktionen in Red Hat Advanced Cluster Security basieren auf den gleichen Abstraktionen, sodass Entwicklungs-, Operations- und Sicherheitsteams ein einziges Framework für das Entwickeln, Bereitstellen und Ausführen von Anwendungen verwenden.

Nutzende von Red Hat Advanced Cluster Security können beispielsweise die Metadaten von Kubernetes-Manifestdateien anzeigen, um wertvollen sicherheitsrelevanten Kontext für eine gesamte Anwendung und ihre beabsichtigte Funktion zu entdecken. Containerzentrierte Lösungen hingegen konzentrieren sich auf Docker-Dateien, die nur die Befehle enthalten, die zum Erstellen eines Images aufgerufen werden können.

Red Hat Advanced Cluster Security nutzt auch die Vorteile standardisierter Paketmanagementtools und stellt sie auf dieselbe Weise bereit wie andere containerisierte Anwendungen. Red Hat Advanced Cluster Security wird beispielsweise mithilfe von Helm Charts bereitgestellt, während containerzentrierte Lösungen

auf Paketmanagementoptionen basieren können, die nicht unbedingt Kubernetes-freundlich sind. Durch die Verwendung offener, standardisierter Abstraktionen und Pakete für die Sicherheit werden Unstimmigkeiten zwischen Sicherheitsteams und anderen Stakeholdern beseitigt, was letztlich die Zusammenarbeit optimiert und wertvolle Zeit spart.

Nutzung des Prinzips *„Einmal konfigurieren, in vielen verschiedenen Umgebungen einsetzen“* für eine einfache *Verlagerung nach links*

Kubernetes-native Sicherheit reduziert auch den Aufwand für das Implementieren sicherer Konfigurationen, da Kubernetes deklarative Konfigurationen in der gesamten Umgebung nach dem Prinzip *„Einmal konfigurieren, in vielen verschiedenen Umgebungen einsetzen“* orchestriert. Mit diesem Ansatz können Unternehmen problemlos Hunderte von Prüfungen für Best Practices und Sicherheitsrichtlinien automatisieren.

Mit Red Hat Advanced Cluster Security können Nutzende eine einzige Konfiguration, beispielsweise eine Netzwerkrichtlinie, an sämtliche Pods in einem Deployment propagieren, anstatt Kontrollen auf Systemebene auf den verschiedenen Hosts eines Clusters zu konfigurieren, wie dies bei einigen containerzentrierten Lösungen erforderlich ist. Containerzentrierte Lösungen nutzen die Skalierbarkeit und Resilienz von Kubernetes nur bedingt. Red Hat Advanced Cluster Security nutzt diese Attribute der nativen Kubernetes-Kontrollen zum Vorteil eines Unternehmens.

Die Sicherheit containerisierter Anwendungen wird durch Kontrollen beeinflusst, die entlang der Softwarelieferkette implementiert werden, bevor die Container tatsächlich ausgeführt werden. Dieser Ansatz bietet Entwicklungs- und DevOps-Teams die Möglichkeit, ihr bestehendes Wissen über Kubernetes zu nutzen, um eine Organisation dabei zu unterstützen, die *Sicherheit nach links zu verlagern* und die Denkweise in „Sicherheit als Code“ zu verändern. Mit diesem Konzept sind viele Teams vertraut, denn es ähnelt den heutigen erfolgreichen Infrastructure-as-Code-Methoden.

Die Kubernetes-native Sicherheit ergänzt diese bestehenden Initiativen durch das Integrieren von Funktionen wie die Durchsetzung von Richtlinien während des Deployments unter Verwendung der in Kubernetes integrierten dynamischen Zugangskontrolle. Containerzentrierte Lösungen verwenden proprietäre Architekturen, um Richtlinien zum Zeitpunkt des Deployments durchzusetzen. So haben Entwicklungs- und DevOps-Teams nicht die Möglichkeit, Sicherheit enger mit Kubernetes-bezogenen Toolchains zu verknüpfen.

Durch das Anpassen der Sicherheit an die Abstraktionen und Toolchains, mit denen Entwicklungs- und DevOps-Teams bereits vertraut sind, ermöglicht Red Hat Advanced Cluster Security Unternehmen Zeit- und Kosteneinsparungen beim Umsetzen cloudnativer Sicherheitsstrategien.

Schnellere Analyse und Problembehebung

Wichtige Schlussfolgerung: Kubernetes-native Sicherheitslösungen optimieren Untersuchungen, Reaktionen auf Sicherheitsvorfälle und Risikobeurteilungen, indem sie den umfangreichen Kontext direkt aus Kubernetes erfassen.

Die Sicherheitsteams müssen heutzutage zu viele Aufgaben erledigen, um die benötigten Antworten zu erhalten. Sie arbeiten mit einem komplexen Patchwork von Tools und sind auf manuelle Workflows und benutzerdefinierte Integrationen angewiesen, um sie miteinander zu verbinden. Dies bedeutet oft, dass sie sich durch einen endlosen Strom von Warnmeldungen durcharbeiten müssen; Container können ihre Tätigkeit noch erschweren. Sicherheitsverantwortliche stehen vor großen Herausforderungen beim Schutz cloudnativer Umgebungen, darunter:

- Vorfälle und ihre neuen, nichtlinearen Bedrohungsmuster (Indikatoren für Angriffe und Kompromittierungen) sind über Anwendungskomponenten und verteilte Umgebungen verstreut, da Container dynamisch in verschiedenen Maschinen orchestriert werden.
- Bestehende Tools und Workflows können mit den großen Datenmengen, die durch eine große Anzahl von Containern erzeugt werden, nicht Schritt halten.

- Viele traditionelle Ansätze bei der Reaktion auf Sicherheitsvorfälle, die sich auf die Aufbewahrung von zustandsbehafteten Daten stützen, um einen Angriff zu untersuchen sowie die Techniken und das Vorgehen des Angreifers kennenzulernen, sind aufgrund der kurzlebigen Natur von Containern ineffektiv.

Präzisere Erkennung von Bedrohungen

Der detaillierte Kontext, den Kubernetes-native Sicherheit bietet, kann die Untersuchung von Sicherheitsvorfällen und die anschließende Problembehebung beschleunigen. Containerisierte Anwendungen, insbesondere solche mit Microservice-Architekturen, ermöglichen eine zuverlässigere Erkennung von Bedrohungen, da Anomalien in kleineren Komponenten mit vorhersehbaren Aktivitäten leichter zu beobachten und zu identifizieren sind. Durch die Kombination dieser Erkenntnisse mit zusätzlichem Kontext von Kubernetes können tatsächliche Bedrohungen viel leichter erkannt werden als False Positives.

Der Kontext von Kubernetes enthält folgende Informationen über das gesamte Deployment:

- Welcher Prozess ausgeführt wird
- Ob es Ressourcenbeschränkungen gibt, die verhindern, dass Container ihre Nachbarn beeinträchtigen
- Welche Berechtigungen und Fähigkeiten einzelnen Containern gewährt werden
- Inwieweit das Root-Dateisystem des Containers beschrieben werden kann
- Welche Block Devices, Konfigurationen oder Secrets vorhanden sind

Außerdem enthält der Kontext wertvolle Metadaten: Anhand der Labels lässt sich feststellen, um welche Anwendung es sich handelt, während die Anmerkungen Beschreibungen zur Anwendung hinzufügen.

Red Hat Advanced Cluster Security nutzt diesen Kontext, um relevante Daten in Workloads – Informationen über Container Images und Aktivitäten auf Systemebene von einzelnen Containern zur Runtime – mit Konfigurationsdaten aus Kubernetes abzugleichen. Die Sicherheit von Kubernetes wirkt sich auf die Sicherheit der einzelnen Container aus und umgekehrt.

Red Hat Advanced Cluster Security erfasst Daten zu einzelnen Containern und deren Beziehung zueinander und konzentriert sich auf Sicherheitsprobleme in großen Umgebungen. In containerzentrierten Lösungen werden diese Kubernetes-Metadaten nicht erfasst, sodass die Sicherheitsverantwortlichen nur einen begrenzten Einblick in die zugrunde liegende Ursache des Vorfalls haben.

Eine Kubernetes-native Lösung integriert disparate Daten in Containern, Kubernetes und im Lifecycle der Anwendung – vom Erstellen der Images bis zum Ausführen der Container. Mit diesen Informationen können Teams die Runtime-Aktivität im Vergleich dazu analysieren, für welche Aktivitäten Container basierend auf Kubernetes-Einstellungen deklarativ konfiguriert wurden. Dadurch kann Red Hat Advanced Cluster Security anomale und verdächtige Runtime-Aktivitäten präziser erkennen, False Positives vermeiden und Alarmermüdung reduzieren.

Containerzentrierte Lösungen hingegen stützen sich beim Erkennen von Bedrohungen ausschließlich auf Informationen aus aktiven Containern, was zu einer Flut von Warnmeldungen führt, die die Sicherheitsverantwortlichen bearbeiten müssen. Durch das automatische Zusammentragen bedrohungsrelevanter Informationen im gesamten Lifecycle sowie in Containern und Kubernetes beseitigt Red Hat Advanced Cluster Security Informations-Silos, die Sicherheitsverantwortliche manuell zusammensetzen müssen.

Implementierung einer risikobasierten Sicherheitspriorisierung

Kubernetes-native Sicherheit korreliert auch Informationen über Schwachstellen mit dem Einblick in Konfigurationen und Beobachtungen von Runtime-Aktivitäten, um die Wahrscheinlichkeit einer Ausnutzung bestimmter Schwachstellen zu bewerten.

Die Ausnutzung einer Schwachstelle, die bestimmte Berechtigungen erfordert, hängt zum Beispiel davon ab, ob diese Berechtigungen für Container in einem bestimmten Deployment existieren. Ebenso ist eine Schwachstelle, die durch Schreiben in das Dateisystem eines Containers ausgenutzt wird, möglicherweise nicht besorgniserregend, wenn das Dateisystem schreibgeschützt ist. Und für ein bestimmtes Deployment kann ein hohes Maß an Netzwerkzugriff konfiguriert sein, weil es sich um einen nicht produktiven Lab-Cluster oder eine externe Webanwendung handelt.

Red Hat Advanced Cluster Security ermittelt für die einzelnen Deployments in der Umgebung einen Risikograd und stuft sie für die Sicherheitsverantwortlichen ein. Containerzentrierte Lösungen bewerten Risiken in erster Linie auf der Basis von Image-Schwachstellen und standardisierten Kennzahlen wie dem Common Vulnerability Scoring System (CVSS), wodurch jegliche Art von Risikobewertung ineffektiv wird. Ein ganzheitliches Risikoverständnis lässt sich nur mit einem umfassenden Einblick in Kubernetes erreichen.

Die Analyse von und Reaktion auf Vorfälle ist nur so effektiv wie die Daten, die eine Sicherheitslösung erfasst und zur Untersuchung bereitstellt. Angesichts begrenzter Zeit und Ressourcen benötigen Sicherheitsverantwortliche eine schnellere Analyse und Problembehebung der Kubernetes-nativen Sicherheit.

Minimierung operativer Risiken mit Kubernetes-nativer Sicherheit

Kubernetes ist eine robuste, skalierbare Plattform, auf der nahezu jede Anwendung, einschließlich kritischer Workloads, ausgeführt werden kann. Die Entwicklung von Kubernetes konzentriert sich auf Funktionen, die die operative Resilienz erhöhen, darunter:

- Selbstreparaturfunktion, einschließlich der Fähigkeit, Container-Neustarts durchzuführen
- Umplanung
- Abschalten von Containern, wenn sie die Zustandsprüfung nicht bestehen

Kubernetes ist außerdem hochgradig skalierbar, da es die gleichen Prinzipien anwendet, mit denen Google wöchentlich Milliarden von Containern ausführt. Die Kubernetes-Infrastruktur erfüllt anspruchsvolle operative Anforderungen hinsichtlich Verfügbarkeit (Uptime), Zuverlässigkeit (mittlere Betriebsdauer zwischen Ausfällen), Latenz und anderen kritischen Messgrößen mit direkten betrieblichen Auswirkungen. Kubernetes ermöglicht dies, ohne das operative Risiko für Anwendungen, Infrastruktur oder das Unternehmen zu erhöhen.

Einige Sicherheitstools bieten eine aktive Durchsetzung von Sicherheit, die sich direkt auf eine Umgebung auswirken kann, indem Container oder Prozesse abgeschaltet (oder deren Start verhindert), der Netzwerkverkehr blockiert oder die Systemaufrufe einer Anwendung einschränkt werden. Diese Tools erhöhen das Risiko für das Unternehmen aus 2 Hauptgründen:

1. Irrtümlich durchgeführte Aktionen können zu einem Ausfall der Anwendung führen.
2. Tools werden zu zusätzlichen Abhängigkeiten, um die sich die Operations-Teams kümmern müssen. Wenn sie geschlossen ausfallen, können sie die Abläufe erheblich stören, und wenn sie offen ausfallen, ist das Unternehmen ungeschützt.

Ein Sicherheitstool muss diese Risiken mindern.

Skalierbare Durchsetzung

Wichtige Schlussfolgerung: Kubernetes-native Sicherheit ermöglicht eine hochgradig skalierbare Sicherheitsdurchsetzung bei gleichzeitigem Minimieren potenzieller Unterbrechungen kritischer Anwendungen mithilfe nativer Kubernetes-Sicherheitskontrollen.

Die Nutzung des Orchestrators zum Ausführen von Funktionen zur Sicherheitsdurchsetzung minimiert das operative Risiko und bietet eine höhere Skalierbarkeit in Kubernetes-Umgebungen. Dieser Ansatz war nicht möglich, als Infrastrukturplattformen noch keine nativen Sicherheitsfunktionen hatten. Kubernetes umfasst heute:

- Netzwerkrichtlinien für die Netzwerksegmentierung
- Admission Controller zum Abfangen von Anfragen an den Kubernetes-API-Server
- Secrets zum Speichern sensibler Zugangsdaten
- Role-based Access Control für die Autorisierung von Nutzenden und Service-Konten
- Viele weitere Funktionen

Dieser Architekturansatz macht zusätzliche Sicherheitstools für bestimmte Durchsetzungsfunktionen überflüssig.

Im Gegensatz dazu bergen containerzentrierte Lösungen mehrere operative Risiken für ein Unternehmen. So müssen containerzentrierte Sicherheitslösungen häufig einen Inline-Proxy verwenden, um den Netzwerkverkehr zwischen Containern zu beschränken. Oder sie ergreifen eingriffsintensivere Maßnahmen, wie beispielsweise das Überschreiben von IP-Regeln für Firewalls.

Damit Container nicht mit kritischen Schwachstellen bereitgestellt werden oder nicht zugelassene Pakete enthalten, kann eine containerzentrierte Lösung auf einen Shim zurückgreifen, der Anfragen an die Container Engine abfängt und außer Kraft setzt. Dies führt zu Single Points of Failure in der gesamten Umgebung – wenn ein Inline-Proxy oder ein Container Engine Shim ausfällt, können auch die Anwendungen ausfallen.

Red Hat Advanced Cluster Security speichert Richtlinien für die Netzwerksegmentierung, die Zugangskontrolle und andere Sicherheitsfunktionen in Kubernetes, wodurch operative Abhängigkeiten entfallen. Wenn Red Hat Advanced Cluster Security aus irgendeinem Grund nicht mehr verfügbar ist, werden die Sicherheitsrichtlinien weiterhin durchgesetzt. Bei containerzentrierten Lösungen ist dies nicht der Fall.

Containerzentrierte Lösungen führen auch zu Skalierbarkeitsproblemen, da in einigen Fällen separate Durchsetzungskomponenten auf den einzelnen Hosts innerhalb eines Clusters ausgeführt werden müssen. Dieser Ansatz verursacht einen Performance-Overhead, wirkt sich auf die Ressourcenverfügbarkeit für Anwendungs-Container aus und erfordert zusätzliche Aufmerksamkeit von den Operations-Teams, was die operative Gesamtbelastung eines Unternehmens erhöht.

Mit Red Hat Advanced Cluster Security ist der Orchestrator der wichtigste Punkt für die Sicherheitsdurchsetzung und nutzt die inhärenten Fähigkeiten von Kubernetes für eine hoch skalierbare und zuverlässige Durchsetzung. Cluster-Administrations- und Infrastruktur-Teams können sich auf die Kubernetes-Abläufe konzentrieren, ohne durch Komponenten von Drittanbietern abgelenkt zu werden. Zusätzlich erhalten Unternehmen neue Sicherheitskontrollen durch die Robustheit und den Umfang des Kubernetes-Systems, das von einem IT-Ökosystem aus Hunderten von Unternehmen und einer Community aus Tausenden von Entwicklerinnen und Entwicklern unterstützt wird.

Eine Kubernetes-native Sicherheitslösung ist auch unabhängig von communitybasierten Architekturen. Ein Beispiel dafür ist das Container Network Interface (CNI), eine Standardspezifikation für Netzwerkschnittstellen. Die Netzwerksegmentierungsfunktionen von Red Hat Advanced Cluster Security arbeiten konsistent mit den verschiedenen CNI-Plugins.

Containerzentrierte Lösungen können von bestimmten Plugins abhängig oder mit diesen inkompatibel sein und so die operativen Probleme eines Unternehmens erschweren. Die Kontrollen von Red Hat Advanced Cluster Security sind nicht von bestimmten Tools oder Plugins abhängig, sodass DevOps- und Operations-Teams die Tools verwenden können, die ihren Anforderungen entsprechen.

Beseitigung operativer Konflikte

Wichtige Schlussfolgerung: Kubernetes-native Sicherheit hilft, operative Probleme zu reduzieren, die auf inkonsistente Konfiguration, mangelnde Koordination und Benutzerfehler zurückzuführen sind.

Um operative Risiken zu minimieren, müssen Unternehmen auch inkonsistente Sicherheitskonfigurationen vermeiden, die zu unvorhersehbarem Anwendungsverhalten und Komplexität führen können und somit zu Benutzerfehlern beitragen. Eine Kubernetes-nativ Architektur verringert die Wahrscheinlichkeit derartiger operativer Probleme.

Sicherheits- und DevOps-Teams, die unterschiedliche Tools, Kontrollen und Richtlinien implementieren, verursachen möglicherweise Inkonsistenzen in ihren Sicherheitskonfigurationen. Inkonsistenz kann auch aus Konflikten zwischen Konfigurationen in verschiedenen Schichten des Infrastruktur-Stacks entstehen. So können beispielsweise auf Container-Ebene Einstellungen konfiguriert werden, die mit den auf Kubernetes-Ebene festgelegten Einstellungen in Konflikt stehen.

Mit der Kubernetes-nativen Sicherheit können Entwicklungs-, Operations- und Sicherheitsteams Richtlinien und deren Durchsetzung zentral in Kubernetes anzeigen und verwalten. Mit Red Hat Advanced Cluster Security bleiben Netzwerksegmentierungsrichtlinien und Richtlinien für die Zugangskontrolle in Kubernetes erhalten, um eine einfache Referenz zu ermöglichen.

Containerzentrierte Lösungen zwingen die Nutzenden, Kontrollen mit unterschiedlichen Tools zu implementieren, und führen zu fehlender organisatorischer Koordination. So können beispielsweise DevOps-Nutzende Kubernetes-Netzwerkrichtlinien verwenden, um den zulässigen Netzwerkverkehr für eine Anwendung einzuschränken, während Nutzende des Sicherheitsteams eine proprietäre Firewall verwenden würden. Dies kann zu widersprüchlichen Konfigurationen und Lücken im Schutz führen.

Ebenso können Inkonsistenzen in Kubernetes- und Container-Konfigurationen unerwartete Auswirkungen auf bestehende Anwendungen haben. Containerzentrierte Lösungen können beispielsweise einschränken, welche Aufrufe des Betriebssystems ein bestimmter Container tätigen kann, selbst wenn innerhalb von Kubernetes keine ähnlichen Einschränkungen durch Pod-Sicherheitsrichtlinien konfiguriert sind. Dies hat zur Folge, dass eine Person, die sich mit Kubernetes befasst, möglicherweise nicht versteht, was die Ursache für ein operatives Problem ist, da die entsprechende Konfiguration auf einer anderen Schicht angegeben ist.

Und in Fällen, in denen Beschränkungen in Konflikt stehen, wird der Verkehr möglicherweise nicht wie vorgesehen zugelassen oder eingeschränkt. Beispielsweise erlaubt eine Kubernetes-Netzwerkrichtlinie den Verkehr zwischen Container A und Container B, aber ein containerzentrierter Inline-Proxy schränkt den Verkehr zwischen diesen Containern ein. Die Folge ist ein operatives Chaos.

Kubernetes-native Sicherheit behandelt Kubernetes als Source of Truth für Sicherheitsfunktionen und verringert den Umfang des Kontextwechsels, der von Operations- und SRE-Teams (Site Reliability Engineering) verlangt wird. Sicherheitsprobleme werden direkt den Kubernetes-Objekten und -Ressourcen zugeordnet – Aufgaben, mit denen diese Teams bereits täglich arbeiten, um die Verfügbarkeit und Zuverlässigkeit von Kubernetes zu gewährleisten.

Dieser Ansatz reduziert die operative Komplexität, die zu Benutzerfehlern beiträgt, die sich negativ auf Kubernetes-Umgebungen auswirken. Containerzentrierte Lösungen erfordern, dass Operations- und SRE-Mitarbeitende Sicherheitsprobleme manuell zwischen herstellerspezifischen Informationen und den Daten in Kubernetes abgleichen. Mit Red Hat Advanced Cluster Security können Nutzende die Informationen in einem Format anzeigen, das bereits mit dem von Kubernetes übereinstimmt. So können Cluster-Operatoren Probleme schneller und einfacher lokalisieren und mit ausgewählten Toolchains und Workflows lösen.

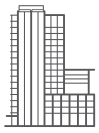
Fazit

Cloudnative Technologien transformieren die Art und Weise, wie Unternehmen ihre Anwendungen ausführen, und transformieren folglich auch ihre Denkweise über das Thema Sicherheit. Sie ermöglichen das Einführen von DevSecOps und anderen Methoden der Zusammenarbeit und führen unsere Branche zu einem Security-as-Code-Modell. Im Zentrum dieses Generationswechsels bei der Infrastruktursoftware steht Kubernetes, das führende Container-Orchestrierungssystem. Es gibt zwar viele Ansätze für den Schutz von Container-Umgebungen, doch die Kubernetes-native Sicherheit bietet einen tieferen Einblick, schnellere Analysen und einfachere Abläufe.

Die Kubernetes-native Sicherheit von Red Hat bietet Unternehmen, die containerisierte Anwendungen entwickeln und ausführen, mehrere besondere Vorteile:

- Erhöhter Schutz durch Beseitigen blinder Flecken und Aufdecken kritischer Schwachstellen und Fehlkonfigurationen, die nur bei Kubernetes auftreten
- Zeit- und Kostenersparnis durch verkürzte Lernprozesse für Teams und beschleunigte Ermittlung und Problembehebung mit nützlichem Kontext von Kubernetes
- Reduziertes operatives Risiko durch die Verwendung von Kubernetes für skalierbare Durchsetzungsfunktionen und die Beseitigung operativer Komplexität, die durch inkonsistente Konfigurationen und Benutzerfehler entsteht
- Standardisierte Plattform für Use Cases für Sicherheitslösungen im Lifecycle cloudnativer Anwendungen, die von Entwicklungs-, Operations- und Sicherheitsteams genutzt werden kann

Mit Red Hat Advanced Cluster Security können Unternehmen ihre Kubernetes-Umgebungen standortunabhängig, in der Produktion und in großem Umfang effektiver schützen.



Über Red Hat

Red Hat, weltweit führender Anbieter von Open Source-Softwarelösungen für Unternehmen, folgt einem communitybasierten Ansatz, um zuverlässige und leistungsstarke Linux-, Hybrid Cloud-, Container- und Kubernetes-Technologien bereitzustellen. Red Hat unterstützt Kunden bei der Integration neuer und bestehender IT-Anwendungen, der Entwicklung cloudnativer Applikationen, der Standardisierung auf unserem branchenführenden Betriebssystem sowie der Automatisierung, Sicherung und Verwaltung komplexer Umgebungen. Dank der vielfach ausgezeichneten Support-, Trainings- und Consulting-Services ist Red Hat ein bewährter Partner der Fortune 500-Unternehmen. Als strategischer Partner von Cloud-Providern, Systemintegratoren, Applikationsanbietern, Kunden und Open Source Communities unterstützt Red Hat Unternehmen auf ihrem Weg in die digitale Zukunft.

facebook.com/redhatinc
@RedHatDACH
linkedin.com/company/red-hat

EUROPA, NAHOST, UND AFRIKA (EMEA)
00800 7334 2835
de.redhat.com
europe@redhat.com

TÜRKEI
00800 448820640

ISRAEL
1 809 449548

VAE
8000-4449549

de.redhat.com
#F28129_0421