Beim Knoten angefangen: Ihr umfassender Leitfaden zu Kubernetes-Sicherheit

mit Prisma Cloud

paloalto NETWORKS | PRISMA CLOUD BY PALO ALTO NETWORKS

Inhalt

Einführung

Die meisten Diskussionen über Kubernetes®-Sicherheit drehen sich darum, wie schwierig es ist, die Cluster zu schützen. Es heißt, da Kubernetes nur eine Handvoll nativer Sicherheitsfunktionen biete, sei es ausgesprochen schwierig, jede Ebene einer Kubernetes-Umgebung zu schützen.

Es stimmt, dass nur wenige Sicherheitstools in Kubernetes integriert sind und dass verschiedene Arten potenzieller Schwachstellen über mehrere Infrastrukturebenen hinweg berücksichtigt werden müssen, um Kubernetes zu schützen. Das heißt jedoch nicht, dass man sich Kubernetes-Sicherheit als hoffnungslos kompliziert vorstellen muss.

Ganz im Gegenteil: Es ist von Vorteil, dass Kubernetes eine so ausgedehnte Plattform mit so vielen Integrationen ist, denn dadurch ist es einfach, einen systematischen Satz automatisierter Prozesse aufzubauen, der die Sicherheit zu einem zentralen Element des Entwicklungs- und Implementierungsprozesses von Kubernetes macht. Das Ergebnis ist eine stark integrierte Sicherheitsstrategie, die Bedrohungen auf allen Ebenen Ihres Stacks abwehrt.

In diesem E-Book wird erläutert, wie Sie eine Sicherheitsstrategie entwickeln, die Ihre übrigen Prozesse auf der Basis von Kubernetes verstärkt und nicht behindert. Sicherheitsherausforderungen mit Kubernetes werden beim Knoten beginnend identifiziert und für jede werden spezifische Lösungen vorgestellt, wobei der Schwerpunkt auf automatisierten, skalierbaren Ansätzen liegt, mit denen auf Kubernetes basierende Workloads geschützt werden, ganz gleich wie groß Ihr Cluster ist oder welche Art Infrastruktur Sie zum Hosten nutzen – On-Premises, öffentliche Cloud oder Managed Services.

1. Kapitel: Grundlagen der Kubernetes-Sicherheit

Bevor wir auf spezifische Sicherheitsherausforderungen in Kubernetes und Strategien zu ihrer Bewältigung eingehen, sollen hier zunächst allgemeine Überlegungen zu Kubernetes-Sicherheit angestellt werden.

Kubernetes ist vielschichtig

Zuallererst muss man sich vor Augen führen, dass Kubernetes eine komplexe Plattform ist, die aus über einem halben Dutzend unterschiedlicher Komponenten besteht. Neben einem API-Server für die Kommunikation zwischen unterschiedlichen Teilen eines Clusters enthält sie einen Aufgabenplaner, der für die Verteilung der Workloads verantwortlich ist, und Controller, die den Status von Kubernetes selbst verwalten. Außerdem umfasst Kubernetes einen Agenten, der auf jedem Knoten oder Server innerhalb eines Clusters läuft, und einen Speicher für Schlüssel-Wert-Paare, der die Clusterkonfiguration enthält.

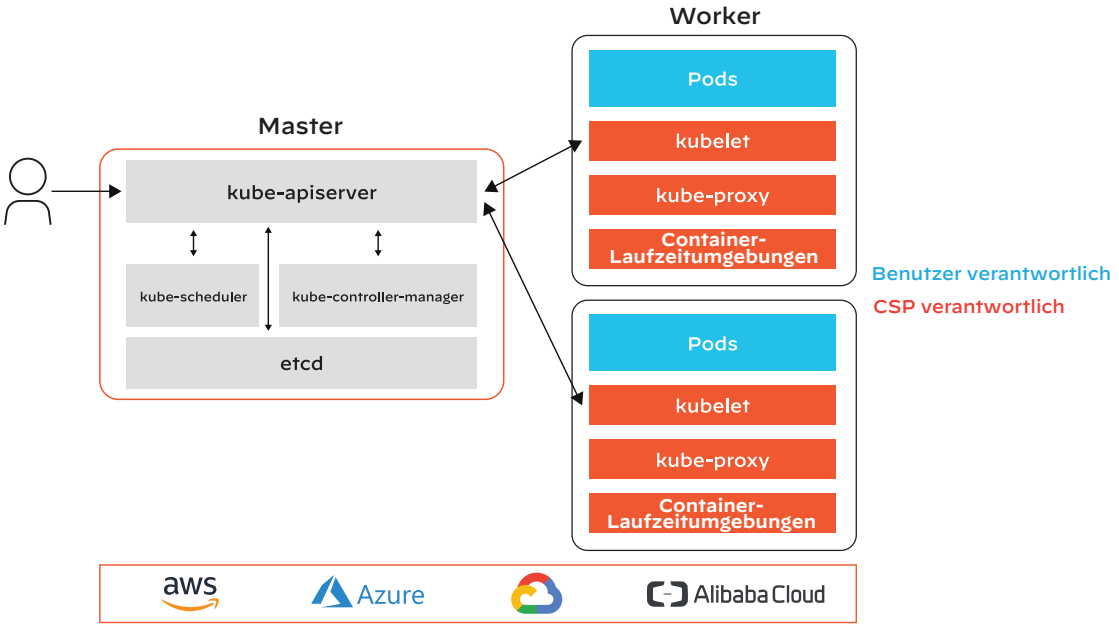

Abbildung 1: Die Architektur verwalteter Kubernetes-Services

Das sind nur die Hauptkomponenten von Kubernetes selbst. Ein funktionelles Cluster besteht aus vielen komplexen, miteinander verzahnten Komponenten, darunter einer Container-Laufzeitumgebung zum Ausführen von Containern, einem persistenten Speicher, einem Protokollierungstool, den Betriebssystemen der einzelnen Knoten usw.

Jede dieser Komponenten hat möglicherweise ihre eigenen Schwachstellen. So könnten Container-Laufzeitumgebungen beispielsweise aufgrund von Programmierfehlern die Ausweitung der Zugriffsrechte innerhalb eines Containers gestatten. Der Kubernetes-API-Server könnte inkorrekt konfiguriert sein, sodass Angreifer auf Ressourcen zugreifen können, die nicht zugänglich sein sollten. Containerisierte Anwendungen oder die auf Kubernetes-Knoten ausgeführten Betriebssysteme könnten Schwachstellen enthalten, die Angriffe zur Ausweitung der Zugriffsrechte oder Zugriff auf sensible Daten ermöglichen. Die Liste der Beispiele ließe sich fortsetzen.

Kurz gesagt, ist zum Schutz von Kubernetes der Schutz einer großen Vielfalt unterschiedlicher Komponenten erforderlich, die jeweils ganz eigene Sicherheitsanforderungen haben. Es gibt nicht den einen Satz von Tools oder Prozessen, die alle Aspekte eines Kubernetes-Clusters mühelos vor allen erdenklichen Bedrohungen schützen. Auch die Abwehr muss vielschichtig sein.

Native Schutzmechanismen von Kubernetes

Eine weitere Komplikation in puncto Sicherheit ist der Umstand, dass Kubernetes trotz seiner integrierten Sicherheitsfunktionen kaum dazu in der Lage ist, sich ohne Unterstützung durch externe Tools selbst zu schützen.

Administratoren können in Kubernetes Richtlinien für die rollenbasierte Zugriffskontrolle (RBAC) definieren, um den unbefugten Zugriff auf Cluster-Ressourcen zu verhindern. Sie können auch Pod-Sicherheitsrichtlinien und Netzwerkrichtlinien konfigurieren, um bestimmte Arten des Missbrauchs auf Pods und im diese verknüpfenden Netzwerk zu unterbinden. Zudem können sie Ressourcenquoten festlegen, um im Falle eines erfolgreichen Angriffs den Schaden auf den direkt betroffenen Teil des Clusters zu begrenzen. Damit lassen sich Denial-of-Service-Angriffe entschärfen, weil die Angreifer nur die Ressourcen im infiltrierten Teil des Clusters für sich beanspruchen können (zumindest, solange sie sich nicht weiter im Cluster ausbreiten können).

Administratoren können in Kubernetes Richtlinien für die rollenbasierte Zugriffskontrolle (RBAC) definieren, um den unbefugten Zugriff auf Cluster-Ressourcen zu verhindern.

Diese und ähnliche native Sicherheitsfunktionen von Kubernetes schließen bestimmte Sicherheitslücken innerhalb eines Kubernetes-Clusters. Gegen viele andere Sicherheitsrisiken, etwa Exploits, die auf das Betriebssystem von Knoten oder Container-Laufzeitumgebungen abzielen, können sie allerdings wenig ausrichten.

Für eine ganzheitliche Sicherheitsstrategie in Kubernetes muss man über die Handvoll integrierter Sicherheitsfunktionen in Kubernetes hinausgehen. Letztere können und sollten verwendet werden, wo sie zur Minderung von Sicherheitsrisiken geeignet sind, doch für sich allein bieten sie nicht einmal annähernd alle zum Schutz eines Clusters erforderlichen Funktionen.

In den folgenden Kapiteln beschäftigen wir uns damit, wie eine umfassende Sicherheitsstrategie für Kubernetes umgesetzt werden kann, die über die begrenzten integrierten Sicherheitsfunktionen hinausgeht.

2. Kapitel: Schutz der Kubernetes-Infrastruktur

Die erste übergreifende Ebene, die in einer Umgebung auf Basis von Kubernetes geschützt werden muss, ist die Build-Ebene, d. h. die Tools, mit denen Entwickler Code schreiben, der in einer Kubernetes-Umgebung ausgeführt werden soll.

Diese Tools sind nicht selbst Teil von Kubernetes. Da ein Kubernetes-Cluster jedoch nur so sicher ist wie der Code, der darauf ausgeführt wird, sind Maßnahmen zum Schutz von Code, noch bevor dieser in einem Cluster implementiert wird, eine Voraussetzung für den Schutz aller Aspekte von Kubernetes.

In diesem Kapitel wird erklärt, wie Kubernetes-Builds geschützt werden können, wobei der Schwerpunkt auf den drei Hauptbereichen liegt, in denen Sicherheit in eine automatisierte Entwicklungspipeline integriert werden kann: integrierte Entwicklerumgebungen (IDEs), Konfigurationsmanagement und Continuous Integration.

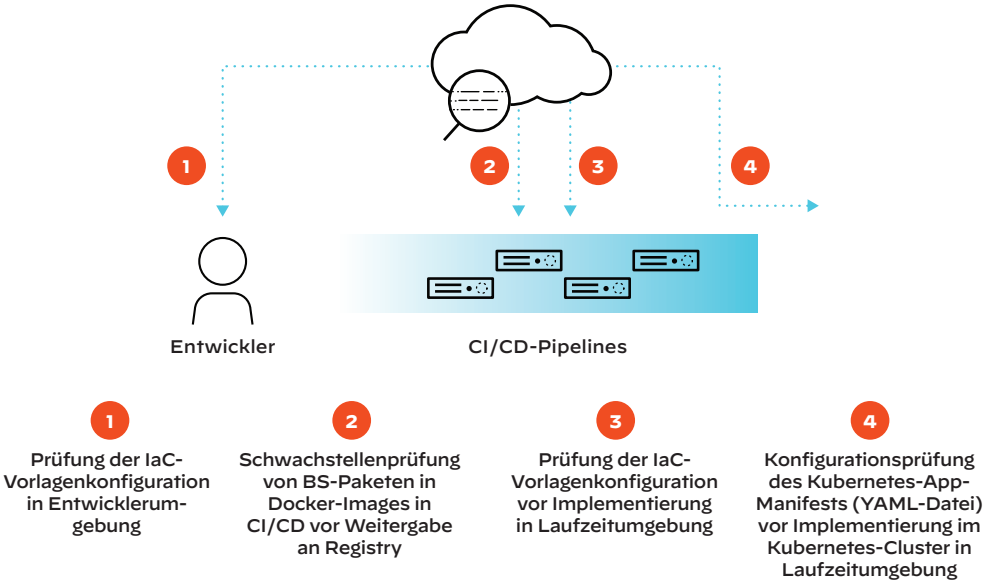

Abbildung 2: Sicherheit in der Entwicklungspipeline

IDEs

Entwickler schreiben den Quellcode für Anwendungen meist in integrierten Entwicklerumgebungen (Integrated Development Environments, IDEs). Da die Anwendungsbereitstellung mit IDEs anfängt, sollte auch die Suche nach Sicherheitslücken bei diesem Tool beginnen. Die meisten IDEs sind mit einer Vielzahl von Schwachstellenscannern diverser Anbieter kompatibel und unterstützen die Suche nach möglichen Sicherheitslücken im Quellcode von Anwendungen mit diesen Scannern.

Continuous Integration (CI)

CI-Tools hosten Quellcode und wandeln ihn in Binärdateien um, die in Kubernetes implementiert werden können. Sie stellen eine weitere Stufe dar, auf der Code auf Schwachstellen untersucht werden sollte. Wie IDEs sind auch CI-Server mit zahlreichen Schwachstellenscannern kompatibel.

Konfigurationsmanagement

Heutzutage stützen sich die meisten Pipelines für die Entwicklung und Implementierung von Kubernetes-Anwendungen auf automatisiertes, richtlinienbasiertes Konfigurationsmanagement in Form von IaC- (Infrastructure-as-Code) und YAML-Dateien. Diese Ansätze ermöglichen es Kubernetes-Administratoren, mittels Code zu definieren, wie ein Cluster (und die Infrastruktur, in der es gehostet wird) konfiguriert sein muss, und diesen Code dann automatisch anzuwenden.

Zusätzlich zur Optimierung des Prozesses zur Bereitstellung einer Kubernetes-Umgebung bieten Konfigurationsmanagementtools eine Gelegenheit, Konfigurationsdateien vor ihrer Anwendung auf Sicherheitsprobleme zu untersuchen. Tools wie Prisma™ Cloud können dies automatisch erledigen, indem sie Ihre IaC- und YAML-Dateien mit Mustern vergleichen, die bekanntermaßen sicher sind. Prisma Cloud lässt sich direkt mit Ihrem Managementsystem für Quellcode, wie GitHub® oder GitLab®, verknüpfen. Das erleichtert die Einrichtung eines vollständig automatisierten Prozesses zum Schutz der Kubernetes-Konfigurationsdateien, der mit den bestehenden Entwicklungspipelines funktioniert.

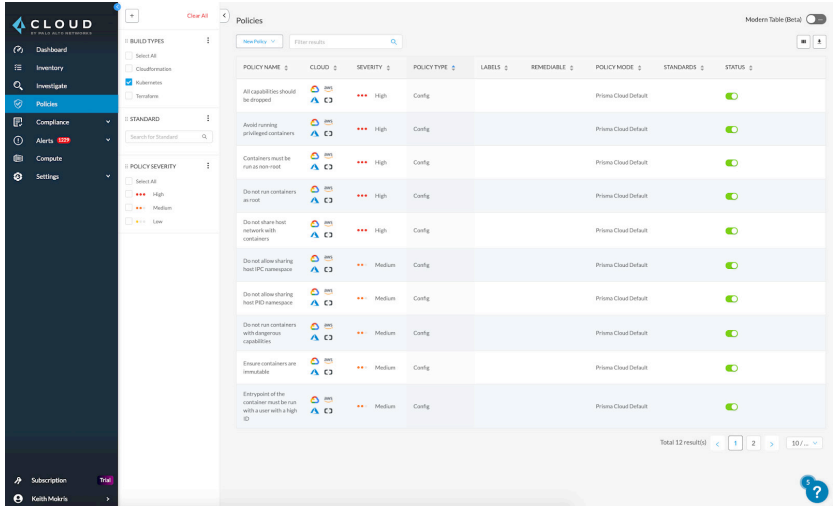

Abbildung 3: Kubernetes-Richtlinien in Prisma Cloud

3. Kapitel: Schutz von Container-Images zur Ausführung auf Kubernetes

In den meisten Fällen werden Anwendungen auf Kubernetes als Container-Images implementiert. (Die Verwaltung anderer Arten von Bereitstellungsobjekten, einschließlich virtueller Maschinen, ist auf Kubernetes möglich, aber weniger üblich.) Container-Images werden auf Schwachstellen überprüft, die im Container-Code selbst oder in jeglichen vorgelagerten Abhängigkeiten bestehen können, auf denen das Image basiert.

Entwickler-Desktop

Es gibt zwei Möglichkeiten, Container-Images auf Sicherheitslücken zu überprüfen. Die eine besteht darin, einzelne Images mit einem Tool wie twistcli manuell zu untersuchen. Dies ist praktisch, wenn Sie ein Image nur einmal prüfen müssen.

Wenn Container-Images mit automatisierten Workflows erstellt werden, müssen Teams wahrscheinlich die Prüfung auf Schwachstellen und Compliance integrieren.

Continuous Integration (CI)

Wenn Container-Images mit automatisierten Workflows auf Plattformen wie Jenkins®, CircleCI® oder Azure® DevOps erstellt werden, müssen Entwickler und DevOps-Teams wahrscheinlich die Prüfung auf Schwachstellen und Compliance in diese Workflows integrieren. Sicherheitsplattformen wie Prisma Cloud können diese Container-Images scannen und mit Frameworks wie Docker CIS Benchmark abgleichen, um Schwachstellen zu ermitteln und Standards auf der Basis von Unternehmens- oder Anwendungsanforderungen durchzusetzen.

Container-Registrys

Zur automatisierten, skalierbaren Überprüfung von Container-Images sollten Sie jedoch regelmäßig alle Images in einer Container-Registry scannen. Registrys sind Repositorys, in denen Container-Images gespeichert sind. Durch die Einbindung eines Schwachstellenscanners in Ihre Registry finden Sie alle Bedrohungen in den in dieser Registry gespeicherten Container-Images.

Eine Herausforderung bei der Überprüfung von Container-Registrys ist, dass es eine Reihe unterschiedlicher Container-Registrys gibt. Manche Kubernetes-Distributionen, etwa Red Hat® OpenShift® und in öffentlichen Clouds gehostete verwaltete Kubernetes-Services, haben eigene integrierte Registrys. Bei anderen können Administratoren aus verschiedenen Registrys anderer Anbieter auswählen.

Wegen der vielfältigen Optionen und Konfigurationen von Registrys ist es wichtig, bei der Wahl eines Tools zur Überprüfung von Container-Images darauf zu achten, dass es mit allen Arten von Registry einsetzbar ist. Prisma Cloud bietet diese Flexibilität und gibt Administratoren somit eine Lösung an die Hand, mit der sie – unabhängig von der Konfiguration ihres Kubernetes-Clusters – alle Images scannen können.

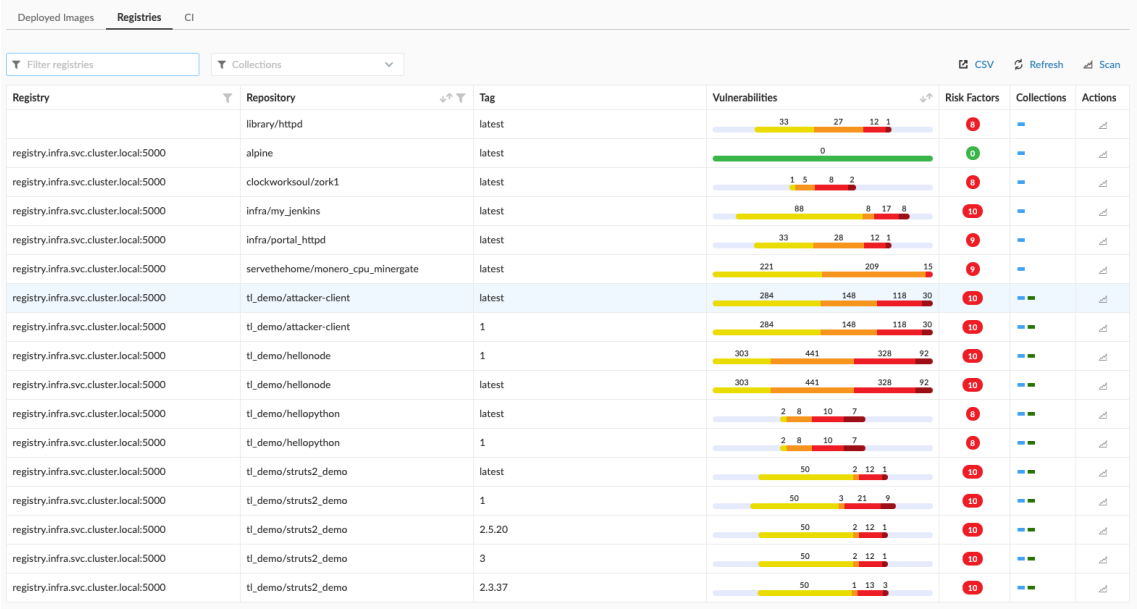

Abbildung 4: Ergebnisse der Überprüfung einer Registry für Container-Images in Prisma Cloud

4. Kapitel: Kubernetes-Laufzeitschutz

Der Schutz von Anwendungen nach der Bereitstellung in einem Cluster zählt zu den kompliziertesten Aspekten der Kubernetes-Sicherheit. Dies liegt daran, dass es so viele verschiedene Schwachstellen gibt, die eine laufende Anwendung betreffen können, und diese Schwachstellen können sowohl über die Anwendung selbst als auch durch Kubernetes ausgenutzt werden.

Neben der Sicherung von Anwendungen vor der Implementierung, wie im vorangegangenen Kapitel erläutert, gibt es auch verschiedene Möglichkeiten, die Sicherheitsrisiken einer Anwendung nach deren Implementierung zu reduzieren.

Transparenz

Vor allem ist es wichtig, kontinuierlich den Überblick über Ihre Kubernetes-Services und -Ressourcen zu behalten. Sicherheitsverstöße können auf vielerlei Weise erfolgen. Je mehr Daten Sie über die Anwendungsumgebung sammeln, desto besser stehen Ihre Chancen, Anomalien zu erkennen, die auf Verstöße hinweisen.

Sicherheitsteams wissen möglicherweise nicht, wo sich alle ihre Cluster befinden, wodurch ihnen ein zusammenhängendes Bild ihres allgemeinen Sicherheitsniveaus fehlt. Prisma Cloud nutzt API-Daten der Anbieter öffentlicher Clouds (CSPs), um Sicherheitsteams zusätzlich zum Konfigurations- und Compliance-Status einen kontinuierlichen Überblick über die Standorte der Cluster zu geben.

Laufzeitschutz

Die Erfassung der Umgebungsdaten von Kubernetes ist relativ einfach. Eine wesentliche Herausforderung beim Monitoring dieser Daten zur gezielten Suche nach Anomalien und möglichen Sicherheitsproblemen ist allerdings, dass sich Kubernetes-Cluster laufend verändern, weil Knoten ausfallen oder heruntergefahren werden, Anwendungen zur Anpassung an Nachfrageschwankungen nach oben oder unten skaliert werden usw. Dadurch ist es praktisch unmöglich, einen Referenzwert für „normale“ Aktivität zu ermitteln, gegen den Anomalien festgestellt werden könnten.

Erwägen Sie stattdessen eine Lösung für Kubernetes-Laufzeitschutz wie Prisma Cloud, die automatisch lernt, wie sich in Kubernetes bereitgestellte Anwendungen unter verschiedenen Bedingungen verhalten. Auf dieser Basis können Benutzer normale Veränderungen im Anwendungsverhalten effektiv von solchen unterscheiden, die auf ein Sicherheitsproblem zurückgehen.

Netzwerksicherheit

Netzwerkbasierte Sicherheitsbedrohungen können Kubernetes auf zwei Arten betreffen: über öffentlich zugängliche Netzwerke, die Anwendungen mit dem Internet verbinden, und über interne Netzwerke, mit denen Kubernetes-Container untereinander Daten austauschen.

Die Erkennung schädlicher Aktivitäten auf beiden Netzwerkarten ist daher unverzichtbar für den Schutz von Kubernetes-Netzwerkressourcen. Da die Netzwerkaktivität wie der Rest einer Kubernetes-Umgebung oder auch die IP-Adressen der Container laufenden Schwankungen unterliegt, ist ein Netzwerkscanner vonnöten, der Container erkennt und die Feinheiten des Netzwerkdatenverkehrs in einem Kubernetes-Cluster versteht. Außerdem benötigen Sie eine Firewall, in der Sie Regeln zum Schutz gegen Bedrohungen aus dem Netzwerk definieren können und die Sie dann entweder benachrichtigt oder bei Regelverstößen Bedrohungen automatisch blockiert. Prisma Cloud enthält Funktionen für Container-sensitive Netzwerkscans und eine Layer-4-Firewall, die all diese Anforderungen erfüllen.

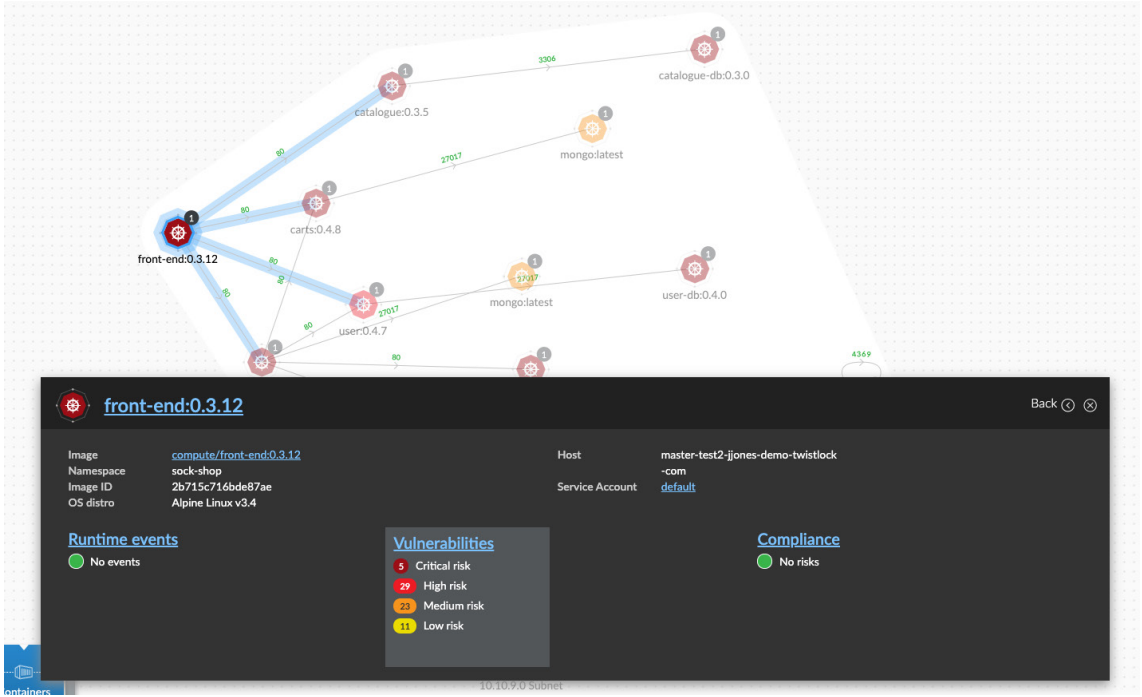

Abbildung 5: Netzwerktopologie und Visualisierung der Containersicherheit in Prisma Cloud

Überwachung der Knoten-Betriebssysteme

Ein Angreifer, der Kontrolle über das Betriebssystem eines Knotens in Ihrem Kubernetes-Cluster erlangt, kann großen Schaden anrichten. Deshalb müssen nicht nur die internen Komponenten von Kubernetes, sondern auch das Betriebssystem eines jeden Knotens unbedingt überwacht werden.

Im Idealfall können Sie dazu dieselbe Monitoring-Lösung nutzen, mit der Sie die Aktivität Ihrer Kubernetes-Anwendungen verfolgen. Dann müssen Sie nicht mit verschiedenen Tools hantieren und mehrere Dashboards im Blick behalten, um von internen und externen Quellen ausgehende Bedrohungen zu erkennen. Prisma Cloud bietet umfassende Überwachungsfunktionen, die alle Arten von Betriebssystem und Cloud-Infrastruktur sowie Kubernetes abdecken.

Sicherheitsaudits und Compliance in Kubernetes

Nicht zuletzt sollten Sie einen Audit-Prozess einrichten, der regelmäßig alle Ebenen Ihres Kubernetes-Clusters und der Konfigurationen prüft, um sicherzustellen, dass Branchenstandards und Best Practices eingehalten

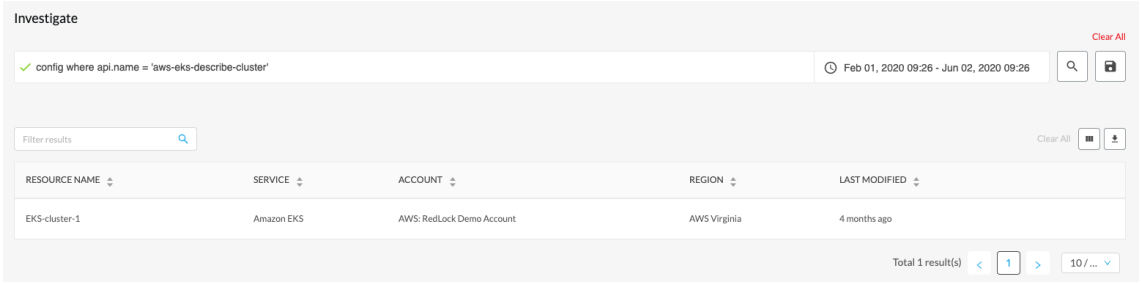

Abbildung 6: Detail einer Kubernetes-Cluster-Untersuchung in Prisma Cloud

werden. Audits erkennen Bedrohungen zwar nicht notwendigerweise in Echtzeit, aber sie helfen Ihnen dabei, Sicherheitsprobleme oder Fehlkonfigurationen, die Sie möglicherweise übersehen haben und die Angreifern ein Einfallstor in Ihr Cluster oder Ihre Anwendungen bieten könnten, frühzeitig zu bemerken.

Prisma Cloud ermöglicht Top-to-Bottom-Sicherheitsaudits von Kubernetes, die alle Komponenten des Clusters auf Abweichungen von festgelegten Benchmarks und Best Practices, etwa dem CIS-Benchmark für Kubernetes, überprüft. Es umfasst über 100 integrierte, konfigurierbare Prüfungen von Konfigurationen, Kommunikation und mehr, die für jede Anwendung und Umgebung angepasst werden können.

Sie können diese Prüfungen dann mit vorkonfigurierten Compliance-Vorlagen für gängige Regularien wie PCI DSS, HIPAA, DSGVO und NIST SP 800-190 verknüpfen. Mithilfe cloudnativer Technologien wie Open Policy Agent oder Kubernetes AuditSink können Sie auch Ihre eigenen Compliance-Prüfungen erstellen. Dies kann nützlich sein, wenn Sie branchenspezifischen Compliance-Regeln unterliegen oder wenn Sie eine maßgeschneiderte Anwendung für einen Geschäftsbereich mit Sicherheitsanforderungen implementieren, die über die allgemeinen Audit-Richtlinien hinausgehen.

Fazit

Kubernetes-Sicherheit mag auf den ersten Blick einschüchtern, nicht zuletzt, weil es so viele verschiedene miteinander verzahnte Komponenten gibt, die geschützt werden müssen. Es ist jedoch möglich, mit einem zentralen Satz Tools die diversen Aspekte der Kubernetes-Sicherheit zu bewältigen. Native Sicherheitsfunktionen in Kubernetes sind ein Teil der Lösung, aber es ist wichtig, auch externe Tools einzusetzen, die Schwachstellen wie Schadcode in Container-Images beheben, Kubernetes-Konfigurationen auf Sicherheitsrisiken überprüfen und eine laufende Überwachung zur Bedrohungserkennung in Echtzeit bieten können.

Prisma Cloud stellt die umfassende Funktionalität bereit, die Sicherheitsteams brauchen, um allen Aspekten der Kubernetes-Sicherheit gerecht zu werden. Es lässt sich auch nativ mit Kubernetes und einer Reihe zugehöriger Tools verknüpfen, was es erleichtert, Sicherheit in die Prozesse für die Entwicklung, Implementierung und Verwaltung von Kubernetes zu integrieren, über die Ihr Team bereits verfügt.

Weitere Informationen über den Einsatz von Prisma Cloud für die Kubernetes-Sicherheit finden Sie auf unserer Website.

Prisma Cloud von Palo Alto Networks

Prisma™ Cloud ist eine äußerst umfassende cloudnative Sicherheitsplattform (CNSP), die Anwendungen, Daten und cloudnative Technologien mit branchenführenden Sicherheits- und Compliancefunktionen während des gesamten Entwicklungslebenszyklus in Multi-Cloud- und Hybrid-Cloud-Umgebungen schützt.

Mit einem integrierten Ansatz bietet es Sicherheits- und DevOps-Teams das nötige Maß an Flexibilität und versetzt sie in die Lage, effektiv zusammenzuarbeiten und die cloudnative Anwendungsentwicklung zu beschleunigen.

Prisma Cloud wird in cloudnative Architekturen und Toolkits eingebunden, um diese – und den gesamten Anwendungslebenszyklus – umfassend zu schützen und voneinander isolierte Punktlösungen zu ersetzen, was die Einführung eines DevSecOps-Ansatzes und schnellere Reaktionen auf Veränderungen der Sicherheitsanforderungen in cloudnativen Architekturen ermöglicht.

Weitere Informationen finden Sie unter paloaltonetworks.com/prisma/cloud.

Oval Tower, De Entrée 99–197
1101 HE Amsterdam
Niederlande

+31 20 888 1883

www.paloaltonetworks.de