Kubernetes Security Operations Center

Securing the Data Center Lurking Inside Your Data Center



Pick any cloud hosting provider and you will have a wide choice of tools to deploy to monitor, report, and respond to configurations and permissions that violate compliance guardrails. Cloud Security Posture Management (CSPM) and Cloud Infrastructure Entitlements Management (CIEM) are established tools that assess the cloud data center and provide critical security functions. But what about the ephemeral data center inside your data center?

Kubernetes is a container orchestration system, but what it essentially creates is a virtual data center that can grow and shrink exponentially in seconds–hundreds of times a day without any human interaction.

Traditional CSPM tools see Kubernetes as a line item in a scan report while missing the intricacies of the control plane which include network security, secrets management, complex workload configurations, and policy enforcement. There are tools that take snapshots of a Kubernetes environment to audit once or even a few times a day, but how helpful is that when security or compliance events (think crypto mining a few minutes at a time) can surface and disappear in seconds? Regardless of the interval, an auditing solution that relies on polling a Kubernetes cluster or simply scanning container images for CVEs will always have gaps and blind spots that can be exploited.

©KSOC Labs, Inc. 2022

Continuous Cluster Discovery

Our agentless cloud connect plugin continuously monitors your cloud environments for clusters, giving you an accurate inventory to combat cluster sprawl and reduce your container attack surface.

KDR: Kubernetes Detection and Response

Event-Driven Security

A highly ephemeral environment such as Kubernetes requires a security solution that is event-driven. The native Kubernetes API supports subscribing to a stream of real-time events, such as workload configuration changes and RBAC policy updates. These streams are a treasure trove of data needed to determine the state of the Kubernetes environment from moment to moment, and they can be consumed without sacrificing cluster performance or resource consumption.

KSOC runs as a horizontally-scalable Kubernetes deployment and hooks into the event stream from a dedicated namespace in each target cluster. This architecture minimizes distribution complexity and allows KSOC to perform moment-by-moment auditing of the entire environment.

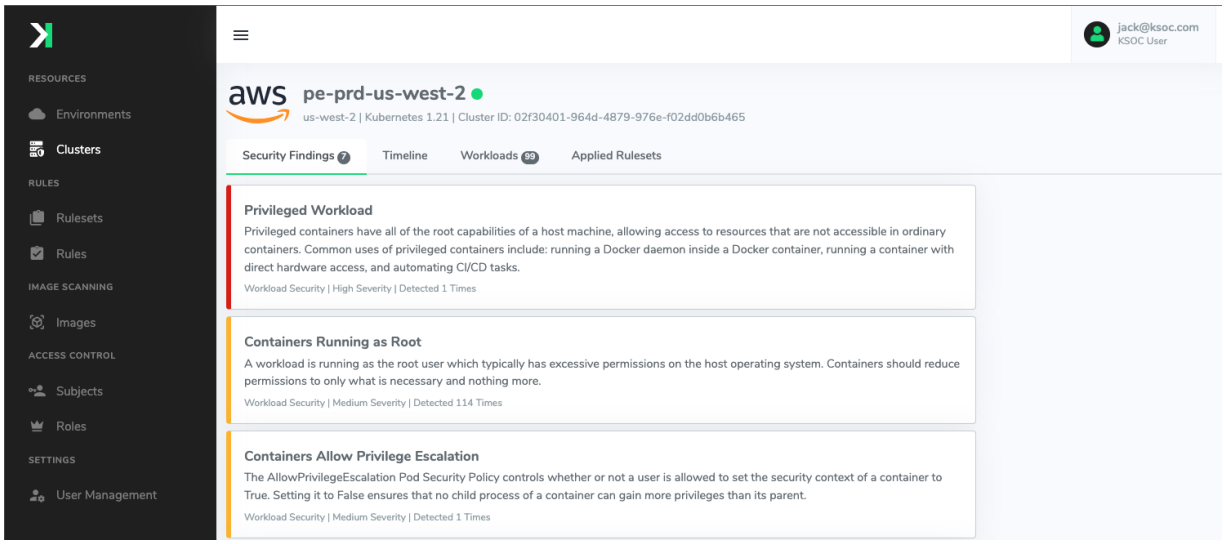
The current state of the environment can be found in the KSOC interface or via the RESTful API.

Real-Time Compliance

KSOC streams Kubernetes API events to the KSOC Analytics backend where they are analyzed by an Open Policy Agent (OPA) rules engine that contains KSOC and/or customer-defined Rego rules to look for security, operational, and compliance violations. KSOC offers CIS, NSA, and OWASP rules out-of-the-box.

The KSOC backend, offered as a SaaS or on-premises deployment, is Kubernetes native and is built to autoscale. This allows KSOC to handle the massive volume generated by Kubernetes API event streams from multiple clusters within multiple Kubernetes environments. Separating the rule processing to run outside the monitored Kubernetes environment minimizes the resource impact on the environment. The evaluation time between when a problematic configuration appears in the Kubernetes environment and when it is identified and appears in KSOC is measured in seconds.

©KSOC Labs, Inc. 2022



Cluster Security Findings: View of the security findings for one cluster.

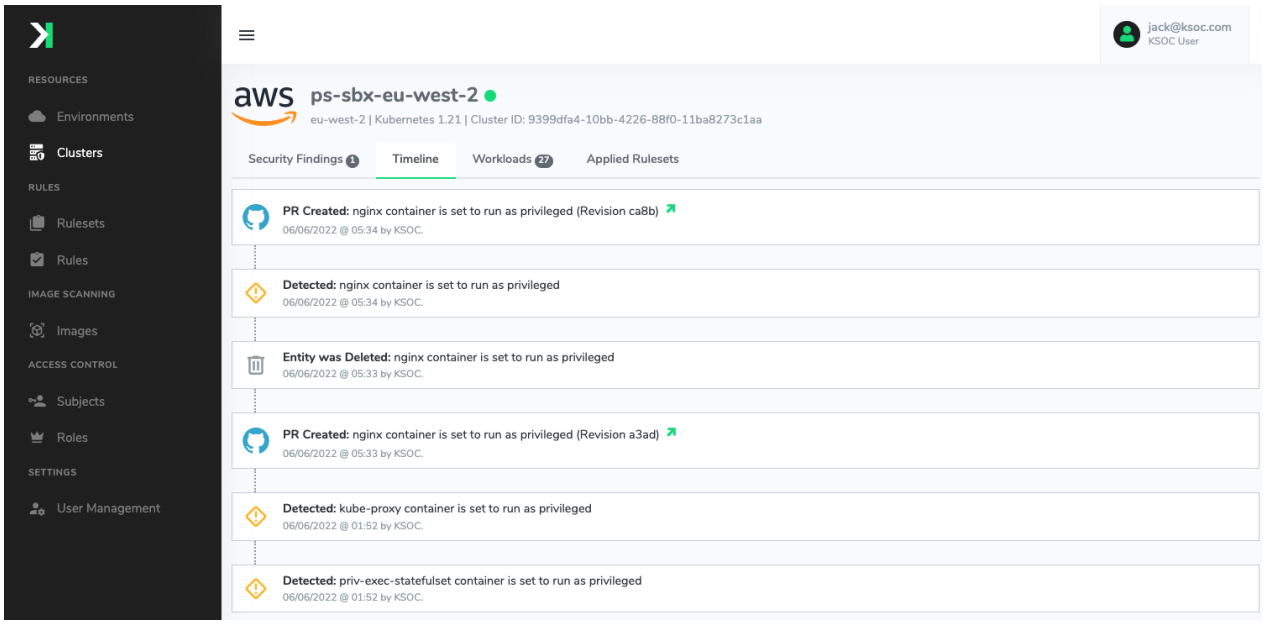
Remediation-as-Code

Identifying violations in real-time is critical, but if all it leads to is a tidal wave of links to KB articles and blogs as a form of remediation, any identified issues will be lost as background noise.

Automatic runtime remediation seems like a logical solution, but making direct changes to a sprawling Kubernetes environment is dangerous and should be looked at with a high degree of skepticism. Automatic remediation of this variety is notoriously difficult to audit and struggles to provide a clear picture of what the full impact of the automatic changes will be.

KSOC's approach to this problem is to codify remediation within the GitOps pipelines themselves by hooking in with the source repositories that back the target Kubernetes environment. KSOC then uses remediation instructions embedded in the Rego rules along with knowledge of these source repositories to simulate building the environment with the suggested fix. This generates a remediation-as-code patch file submitted back to the source repository in a pull request. The pull request contains a diff view between the current and proposed cluster configurations that will run, allowing users to easily review and understand the full impact of incorporating the proposed changes.

Within seconds of accepting the pull request, KSOC will receive an event from Kubernetes indicating the change, re-run the rule that led to the remediation-as-code action, and validate that the security finding has been fixed.



Cluster Timeline: View of the findings timeline for one cluster.

KSCS: Kubernetes Supply Chain Security

Deployed Image Scanning

The vast majority of images inside a container registry have not been used for months or years and will likely not be used in the future. Vulnerability scan results for these unused images create noise for users and make it harder to distinguish the priority for remediation.

KSOC hooks in with your container registries, but rather than scan all images and provide an outsized workload for various teams to investigate, KSOC focuses only on images that are deployed in your clusters. Therefore the resulting list of image vulnerabilities reflects the current state of the Kubernetes environment.

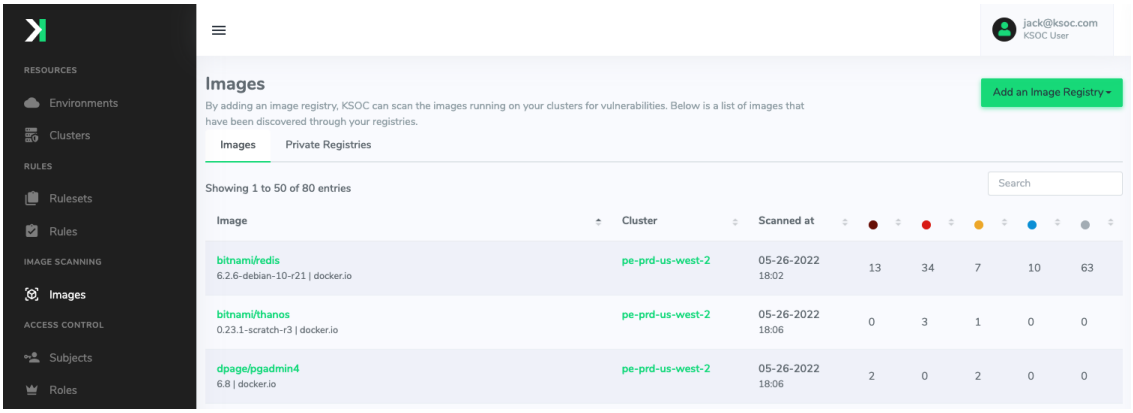


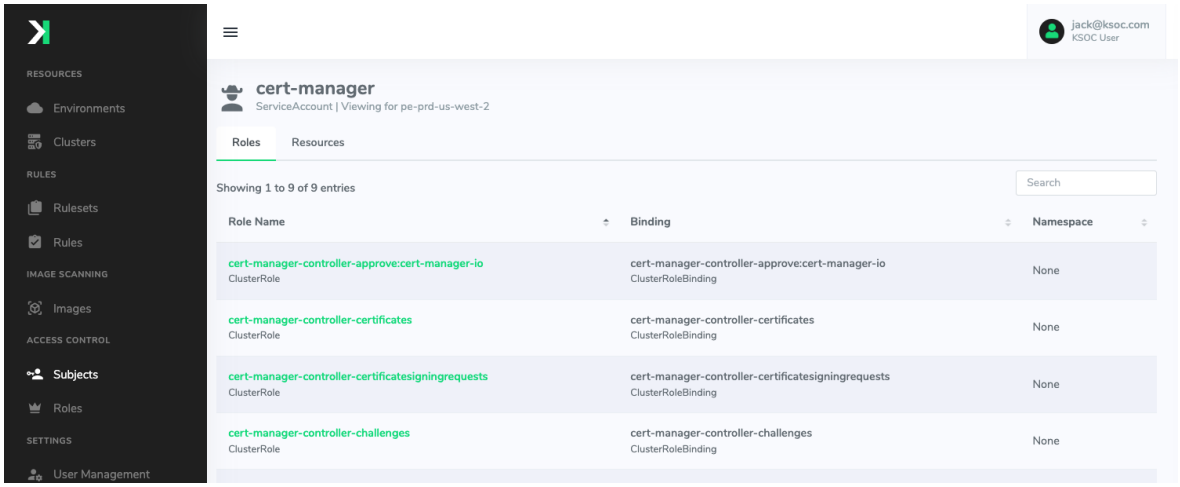
Image List: A view into all currently deployed images and the number of vulnerabilities.

KIEM: Kubernetes Identity and Entitlement Management

RBAC Visibility

Kubernetes RBAC is role centric. Roles and ClusterRoles are first-class objects that are assigned via RoleBindings and ClusterRoleBindings to subjects (users, groups, or service accounts). One of the practical challenges this architecture presents becomes apparent when it is necessary to determine the access level for a specific user or service account. As currently designed, the answer requires iterating over each individual binding to see if the user or service account is bound. This tedious operation becomes exponentially more difficult with each additional cluster in the environment and makes it impossible to understand a subject's access across multiple clusters.

KSOC allows for both role-centric and subject-centric views of a multi-cluster Kubernetes environment. Need to know the current roles that are assigned to user X or service account Y across 20 clusters? The information is only a click or API request away.



Subject Details: View of individual Subject listing all bound Roles and ClusterRoles.