Mitigate Attack Vectors with Alcide for Kubernetes Security

Last month, the Microsoft Azure Security Center published a fully detailed Threat Matrix for Kubernetes. This article identifies attack vectors unique to a Kubernetes environment. This important contribution is derived from the more generalized MITRE ATT&CK® framework that offers a complex matrix of common attack vectors.

As the author of the article, Yossie Weizman, calls it, this is “the first Kubernetes attack matrix: an ATT&CK-like matrix comprising the major techniques that are relevant to container orchestration security, with focus on Kubernetes.”
The release of this matrix is a major milestone in Kubernetes security, helping extend the common language designed by the ATT&CK® matrix to talk about Kubernetes environments and better equipping us to handle threat mitigation in advanced networking.
At Alcide, we’ve created an end-to-end platform that ensures the safety of your Kubernetes environments from development through production.

Take a look at the Azure matrix - showing you in green shading the exact stages where Alcide’s solutions can get you covered:

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Impact
Using Cloud credentials	Exec into container	Backdoor container	Privileged container	Clear container logs	List K8S secrets	Access the K8S API server	Access cloud resources	Data Destruction
Compromised images in registry	bash/cmd inside container	Writable hostPath mount	Cluster-admin binding	Delete K8S events	Mount service principal	Access Kubelet API	Container service account	Resource Hijacking
Kubeconfig file	New container	Kubernetes CronJob	hostPath mount	Pod / container name similarity	Access container service account	Network mapping	Cluster internal networking	Denial of service
Application vulnerability	Application exploit (RCE)		Access cloud resources	Connect from Proxy server	Applications credentials in configuration files	Access Kubernetes dashboard	Applications credentials in configuration files	
Exposed Dashboard	SSH server running inside container					Instance Metadata API	Writable volume mounts on the host	
							Access Kubernetes dashboard	
							Access tiller endpoint	

In this post we'll show you how the Alcide platform helps you cover security by applying best practices throughout the lifecycle and ensuring you leave no doors to your network unlocked. We'll then discuss the new Kubernetes matrix and its mapping to Alcide capabilities.

Best Practices Secure Your Network: The Alcide Platform

Good security through proper Kubernetes configuration and management should be your guiding points when planning deployment of your environment. When asked in the Container Adoption Survey about concerns in Kubernetes deployment, "respondents most frequently cited data security (56%)".
There are a number of methods attackers can employ to stay "under the radar", but by adopting the Alcide Kubernetes Security Platform you can leverage early stage and continuous security assessment, smart detection and prevention to automate monitoring and protect Kubernetes deployments from these invisible threats.
Providing end-to-end support for the entire development lifecycle, the Alcide Kubernetes Security Platform helps you protect your environment based on industry-accepted best practices, including:

- Instant visibility
- Ongoing monitoring, mitigation and detection
- Security at every layer of your deployment: nodes, clusters and pods
- Principle of least privilege

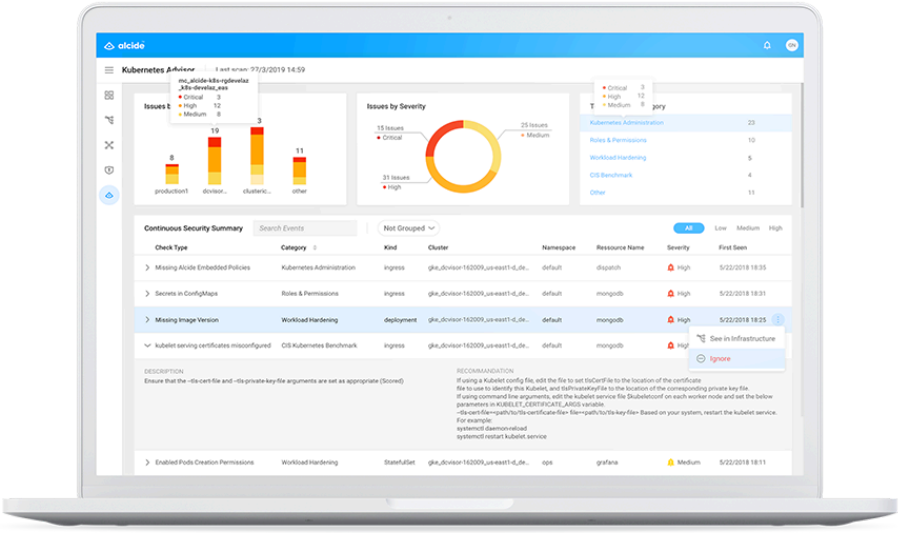
Alcide SaaS platform helps mitigate the threat vectors outlined in the new Kubernetes matrix, helping to identify critical "open doors" and suggesting how to better lock them.

Continuous Audit & Compliance of Kubernetes Clusters with Alcide Advisor

Alcide Kubernetes Advisor is a Kubernetes multi-cluster vulnerability scanner that covers rich Kubernetes and Istio security best practices and compliance checks such as:

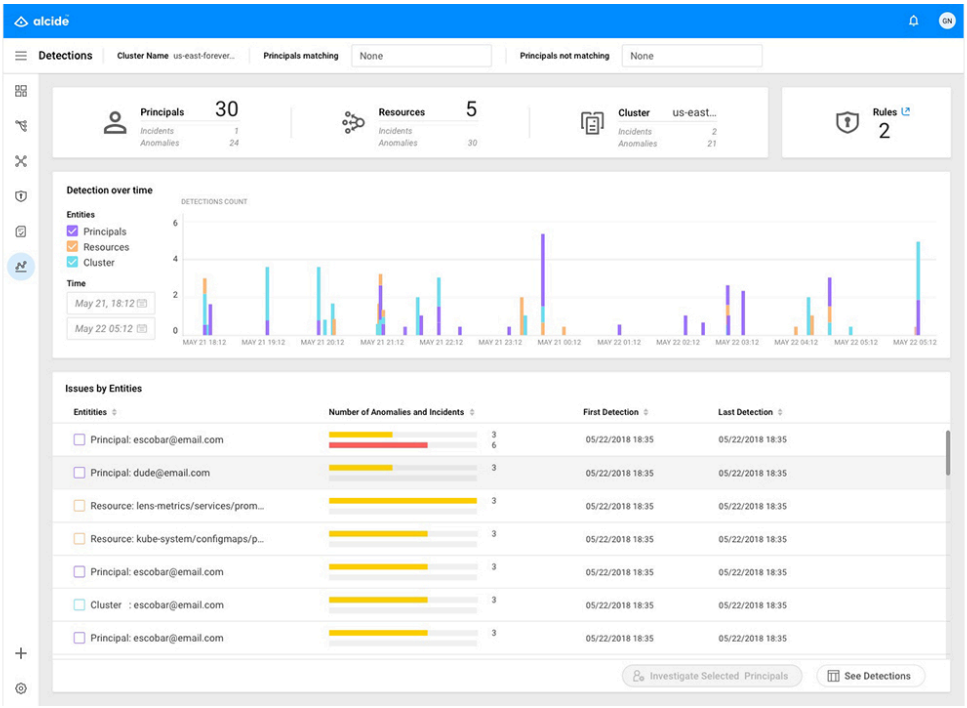
- Kubernetes vulnerability scanning
- Hunting misplaced secrets
- Excessive secret access
- Workload hardening from Pod Security to network policies
- Istio security configuration and best practices
- Ingress controllers for security best practices
- Kubernetes API server access privileges

- Kubernetes operators security best practices



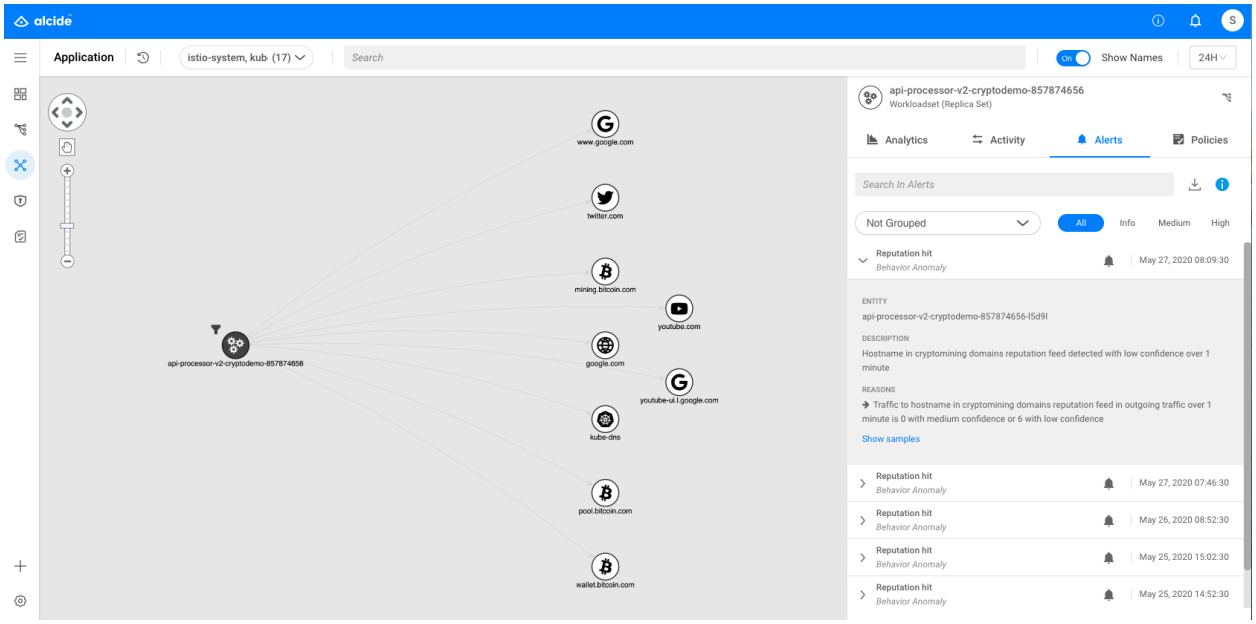
Detect and Analyze Kubernetes Incidents with Alcide kAudit

Ongoing monitoring is vital for your k8s safety.
Alcide kAudit analyzes your Kubernetes logs, identifying abnormal API and administrative activity and compromised k8s resources. Results are then displayed clearly from the advanced dashboard. This enables deep data-based investigation, helping you identify problems and take action immediately without having to sift through raw logs on your own.



Kubernetes Runtime Security with Alcide Runtime

Alcide Runtime (ART) examines workload conformance and automates detection of anomalous behavior. By gathering information about workload behavior and network usage, and processing that data with the use of expert machine learning techniques, ART highlights unexpected usage patterns and unusual data transfers. These indicators let an operator zero-in on potentially compromised or infected workloads, which may then be isolated or, after an investigation, terminated. In addition, Alcide ART policies can be used to guide you in implementing preventative measures including segregation and isolation, helping you minimize the blast radius in case of an attack.



Mapping the K8s Matrix to Alcide Capabilities

The threat vector categories for the Kubernetes (and the Mitre Att&ck) matrix include:

- Initial access
- Execution and Persistence
- Privilege escalation
- Defense evasion
- Credential access
- Discovery
- Lateral movement
- Impact

The following table maps these categories on a high-level to the Alcide modules and capabilities (Advisor, kAudit, Runtime).

<table>
<tr><th colspan="2">Initial Access</th></tr>
<tr><td colspan="2">The initial access tactic consists of techniques that are used for gaining access to the resource. In containerized environments, those techniques enable first access to the cluster. This access can be achieved directly via the cluster management layer or, alternatively, by gaining access to a malicious or vulnerable resource that is deployed on the cluster</td></tr>
<tr><td>Using cloud credentials</td><td rowspan="5">Alcide protects your system from initial access vectors by:
Detecting account takeover and compromised resources (abnormal API activity)
Validating authorized registries and whitelisted container processes as well as public-facing pods
Validating protection of the k8s dashboard</td></tr>
<tr><td>Compromised images in registry</td></tr>
<tr><td>Kubeconfig file</td></tr>
<tr><td>Vulnerable application</td></tr>
<tr><td>Exposed dashboard</td></tr>
<tr><th colspan="2">Execution</th></tr>
<tr><td colspan="2">The execution vector refers to methods used to run malicious code inside a Kubernetes cluster once initial access has been gained.</td></tr>
<tr><td>Exec into container</td><td rowspan="5">Alcide leverages its three modules to help you neutralize the execution vector by:
Validating RBAC configurations for your resources, guiding you by principle of least privilege
Detecting compromised k8s resources
Ensuring attackers cannot deploy new containers or containerized applications
Identifying abnormal administrative and networking activity</td></tr>
<tr><td>bash/command inside container</td></tr>
<tr><td>New container</td></tr>
<tr><td>Application exploit</td></tr>
<tr><td>SSH server running inside a container</td></tr>
<tr><th colspan="2">Persistence</th></tr>
<tr><td colspan="2">Persistence tactics enable access to the cluster to be maintained even if the initial foothold is lost.</td></tr>
<tr><td>Backdoor container</td><td rowspan="3">In addition to the aforementioned capabilities, Alcide also validates immutable root file systems to prevent malicious overwriting.</td></tr>
<tr><td>Writable hostPath mount</td></tr>
<tr><td>Kubernetes CronJob</td></tr>
<tr><th colspan="2">Privilege escalation</th></tr>
<tr><td colspan="2">The privilege escalation tactic consists of techniques that are used by attackers to get higher privileges in the environment than those they currently have.
In containerized environments, this can include getting access to the node from a</td></tr>
</table>

container, gaining higher privileges in the cluster, and even getting access to the cloud resources.	
Privileged container	To help protect against privilege escalation, Alcide continues to offer the monitoring and validation as we've described. You can also proactively identify externally-facing resources and assign firewall policies to promote segregated communication.
Cluster-admin binding	
hostPath mount	
Access cloud resources	
Defense evasion	
Defense evasion is when attackers avoid detection and hide their activity.	
Clear container logs	N/A
Delete Kubernetes events	To prevent defense evasion, continue using Alcide for: its monitoring capabilities, identifying malicious and abnormal activities at all of the layers of your environment and recognizing illicit activity based on policy-based authorization. In addition, reduce risk and exposure by ensuring that Kubernetes cluster resources don't have permissions to create or modify pods or containers using the Kubernetes API server.
Pod / container name similarity	
Connect from proxy server	
Credential access	
The credential access tactic consists of techniques that are used by attackers to steal credentials. In containerized environments, this includes credentials of the running application, identities, secrets stored in the cluster, or cloud credentials.	
List Kubernetes secrets	Keep your secrets secret with Alcide Advisor and Runtime by: ● Hunting for exposed secrets and ensuring they do not find their way into production environments with the Alcide Admission Controller. ● Deploying microservices firewall policies. ● Validating service accounts and preventing activities such as Pod SA Automounts.
Mount service principal	
Access container service account (SA)	
Application credentials in configuration files	
Discovery	
The discovery tactic consists of techniques that are used by attackers to explore the environment to which they gained access. This exploration helps the attackers to perform lateral movement and gain access to additional resources.	
Access the Kubernetes API server	Put discovery to a halt before it starts with Alcide by: ● Identifying abnormal administrative and

Access Kubelet API	network activity and preventing abuse of the k8s API ● Identifying and alerting on scanning attempts ● Verifying access policies for your dashboard ● Ensuring proper isolation of your virtual machines and hosts
Network mapping	
Access Kubernetes dashboard	
Instance Metadata API	
Lateral movement	
Gaining access to various resources in the cluster from a given access to one container, gaining access to the underlying node from a container, or gaining access to the cloud environment.	
Access cloud resources	Continue to leverage Alcide capabilities at the lateral movement vector for all of your environment layers to protect against abnormal and unauthorized activity, implementing our firewall policies, hunting for secrets, validating immutable containers, protecting your dashboard, and detecting compromised resources.
Container service account	
Cluster internal networking	
Applications credentials in configuration files	
Writable volume mounts on the host	
Access Kubernetes dashboard	
Access tiller endpoint	
Impact	
The impact vector is when attackers ultimately have gained enough access to destroy, abuse, or disrupt the normal behavior of the environment.	
Data destruction	Make sure this door stays locked with protection by Alcide, ensuring your data and resources are protected by: ● Identifying attempts to create or modify cluster resources. ● Identifying abnormal activity typically associated with hijacked resources based on our threat intelligence mechanism. ● Enforcing your microservices policies end-to-end, ensuring only authorized and legitimate entities can communicate with one another.
Resource hijacking	
Denial of service	

Conclusion

Deployment of Kubernetes environments is on the rise. At the same time, so are associated attacks and security breaches. For this reason, we need solutions to help proactively protect, mitigate and respond to potential and actual threats throughout the entire development lifecycle. Developers, DevOps and Security teams need a solution that will put their minds to rest - one that will help them quickly analyze and understand activity in their environment and to continuously monitor and protect their resources.
Alcide does just that - the attack vectors described in the new Kubernetes threat matrix are protected end-to-end with security automation from code-to-production security components.