Monitoring Kubernetes sensitive object access

Scenario: Kubernetes is the most used container orchestration platform. It contains sensitive objects within its architecture that, if accessed by an attacker, can lead to further compromise. These objects include configmaps, which are API objects used to store non-confidential data in key-value pairs, and secrets, which store passwords, OAuth tokens, ssh keys, and more. You want to monitor access to Kubernetes cluster sensitive objects and obtain information such as user, group, object, namespace, and authorization reason. The Splunk Security Research team developed this use case to help you detect suspicious requests against Kubernetes sensitive objects.

Prerequisites

To succeed in implementing this use case, you need the following dependencies, resources, and information.

- People: Threat hunter, system administrator, or security tools engineer
- Technologies: Splunk Enterprise or Splunk Cloud Platform
- Data:
  - Common Information Model
  - Kubernetes for AWS, Azure, or GCP

How to use Splunk software for this use case

You can run many searches with Splunk software to monitor Kubernetes sensitive object access. Depending on what information you have available, you might find it useful to identify some or all of the following:

- Suspicious kubectl calls
- Kubernetes accounts accessing sensitive objects
- Kubernetes service accounts with failed or forbidden status

Results

To maximize their benefit, the how-to articles linked in the previous section likely need to tie into existing processes at your organization or become new standard processes. These processes commonly impact success with this use case:

- Kubernetes best practices for securing your cluster

Measuring impact and benefit is critical to assessing the value of monitoring Kubernetes sensitive object access. The following are example metrics that you might want to monitor for reductions when implementing this use case:

- Number of interactions per user group and IP addresses
• User agent types and source user interactions with configmaps or secrets
• Number of service accounts interacting with resources in unusual manner
• Kubectl calls with commands associated with lateral movement
• Kubectl calls creating or accessing resources in an unusual manner

Additional resources

If you have questions about this use case, see the Security Research team’s support options on GitHub. In addition, these Splunk resources might help you understand and implement this use case:

• Blog: Challenges in monitoring Kubernetes environments
• Blog: Approaching Azure Kubernetes security
• Conf Talk: Effective strategies for monitoring Docker and Kubernetes environments
• Conf Talk: Attacking and defending Kubernetes
• Tech Talk: Monitor and troubleshoot Kubernetes-based deployments
• Tech Talk: Splunk Operator for Kubernetes: Effortlessly deploy, scale, and manage Splunk