

UNCLASSIFIED



# **MIRANTIS KUBERNETES ENGINE (MKE) SECURITY TECHNICAL IMPLEMENTATION GUIDE (STIG) OVERVIEW**

**Version 1, Release 1**

**08 May 2024**

**Developed by Mirantis and DISA for the DOD**

UNCLASSIFIED

### **Trademark Information**

Names, products, and services referenced within this document may be the trade names, trademarks, or service marks of their respective owners. References to commercial vendors and their products or services are provided strictly as a convenience to our users, and do not constitute or imply endorsement by the Defense Information Systems Agency (DISA) of any nonfederal entity, event, product, service, or enterprise.

TABLE OF CONTENTS

	Page
<b>1. INTRODUCTION.....</b>	<b>1</b>
1.1 Executive Summary.....	1
1.2 Vulnerability Severity Category Code Definitions.....	1
1.3 STIG Distribution.....	1
1.4 SRG Compliance Reporting.....	2
1.5 Document Revisions.....	2
1.6 Other Considerations.....	2
1.7 Product Approval Disclaimer.....	2
<b>2. ASSESSMENT ASSUMPTIONS.....</b>	<b>4</b>
<b>3. CONCEPTS AND TERMINOLOGY CONVENTIONS.....</b>	<b>5</b>
3.1 MKE Architecture.....	5
<b>4. GENERAL SECURITY REQUIREMENTS.....</b>	<b>7</b>
4.1 Comprehensive Security.....	7

LIST OF TABLES

	Page
Table 1-1: Vulnerability Severity Category Code Definitions .....	1

## LIST OF FIGURES

	<b>Page</b>
Figure 3-1: MKE Architecture.....	6

## 1. INTRODUCTION

### 1.1 Executive Summary

The Mirantis Kubernetes Engine (MKE) Security Technical Implementation Guide (STIG) is published as a tool to improve the security of Department of Defense (DOD) information systems. This document is meant for use in conjunction with other STIGs and appropriate operating system STIGs. MKE is a container orchestration platform for developing and running modern applications at scale, on private clouds, public clouds, and on bare metal. This STIG was tested using MKE version 3.7.

**Orchestration Philosophy:** Whether the application requirements are complex and require medium to large clusters, or simple clusters that can be deployed quickly on development environments, MKE gives users a container orchestration choice. Deploy Kubernetes, Swarm, or both types of clusters and manage them on a single MKE instance or centrally manage the instance using Mirantis Container Cloud.

**Docker Swarm:** Docker Swarm is a native clustering and orchestration solution provided by Docker. It is designed to be simple and user-friendly, making it a good choice for users who are new to container orchestration.

**Kubernetes:** Kubernetes is a more feature-rich and complex container orchestration platform. It is known for its declarative configuration, robust scaling capabilities, and extensive ecosystem of tools and extensions. Kubernetes is widely adopted in large-scale and production environments.

### 1.2 Vulnerability Severity Category Code Definitions

Severity Category Codes (referred to as CAT) are a measure of vulnerabilities used to assess a facility or system security posture. Each security policy specified in this document is assigned a Severity Category Code of CAT I, II, or III.

**Table 1-1: Vulnerability Severity Category Code Definitions**

Category	DISA Category Code Guidelines
CAT I	Any vulnerability, the exploitation of which will <b>directly and immediately</b> result in loss of Confidentiality, Availability, or Integrity.
CAT II	Any vulnerability, the exploitation of which <b>has a potential</b> to result in loss of Confidentiality, Availability, or Integrity.
CAT III	Any vulnerability, the existence of which <b>degrades measures</b> to protect against loss of Confidentiality, Availability, or Integrity.

### 1.3 STIG Distribution

Parties within the DOD and federal government's computing environments can obtain the applicable STIG from the DOD Cyber Exchange website at <https://cyber.mil/>. This site contains

the latest copies of STIGs, SRGs, and other related security information. Those without a Common Access Card (CAC) that has DOD Certificates can obtain the STIG from <https://public.cyber.mil/>.

## 1.4 SRG Compliance Reporting

All technical NIST SP 800-53 requirements were considered while developing this STIG. Requirements that are applicable and configurable will be included in the final STIG. A report marked Controlled Unclassified Information (CUI) will be available for items that did not meet requirements. This report will be available to component authorizing official (AO) personnel for risk assessment purposes by request via email to: [disa.stig\\_spt@mail.mil](mailto:disa.stig_spt@mail.mil).

## 1.5 Document Revisions

Comments or proposed revisions to this document should be sent via email to the following address: [disa.stig\\_spt@mail.mil](mailto:disa.stig_spt@mail.mil). DISA will coordinate all change requests with the relevant DOD organizations before inclusion in this document. Approved changes will be made in accordance with the DISA maintenance release schedule.

## 1.6 Other Considerations

DISA accepts no liability for the consequences of applying specific configuration settings made on the basis of the SRGs/STIGs. It must be noted that the configuration settings specified should be evaluated in a local, representative test environment before implementation in a production environment, especially within large user populations. The extensive variety of environments makes it impossible to test these configuration settings for all potential software configurations.

For some production environments, failure to test before implementation may lead to a loss of required functionality. Evaluating the risks and benefits to a system's particular circumstances and requirements is the system owner's responsibility. The evaluated risks resulting from not applying specified configuration settings must be approved by the responsible AO. Furthermore, DISA implies no warranty that the application of all specified configurations will make a system 100 percent secure.

Security guidance is provided for the DOD. While other agencies and organizations are free to use it, care must be given to ensure that all applicable security guidance is applied at both the device hardening level and the architectural level due to the fact that some settings may not be configurable in environments outside the DOD architecture.

## 1.7 Product Approval Disclaimer

The existence of a STIG does not equate to DOD approval for the procurement or use of a product.

STIGs provide configurable operational security guidance for products being used by the DOD. STIGs, along with vendor confidential documentation, also provide a basis for assessing compliance with cybersecurity controls/control enhancements, which supports system assessment and

authorization (A&A) under the DOD Risk Management Framework (RMF). Department of Defense AOs may request available vendor confidential documentation for a product that has a STIG for product evaluation and RMF purposes from [disa.stig\\_spt@mail.mil](mailto:disa.stig_spt@mail.mil). This documentation is not published for general access to protect the vendor's proprietary information.

AOs have the purview to determine product use/approval in accordance with (IAW) DOD policy and through RMF risk acceptance. Inputs into acquisition or pre-acquisition product selection include such processes as:

- National Information Assurance Partnership (NIAP) evaluation for National Security Systems (NSS) (<https://www.niap-ccevs.org/>) IAW CNSSP #11.
- National Institute of Standards and Technology (NIST) Cryptographic Module Validation Program (CMVP) (<https://csrc.nist.gov/groups/STM/cmvp/>) IAW federal/DOD mandated standards.
- DOD Unified Capabilities (UC) Approved Products List (APL) (<https://www.disa.mil/network-services/ucco>) IAW DODI 8100.04.



## 2. ASSESSMENT ASSUMPTIONS

- The kubectl tool must be installed to perform command line operations for Kubernetes. The software can be found at <https://kubernetes.io/docs/tasks/tools/#kubectl>.
- MKE users must be authenticated using an enterprise authentication system. LDAP is the recommendation and was used to develop this STIG. The process for integrating SAML with LDAP for MKE users is detailed here: <https://docs.mirantis.com/mke/3.7/ops/administer-cluster/configure-ldap-saml-together.html>.
- While most administration of MKE is done through the MKE UI, the Docker CLI is still available. Some checks and fixes for the STIG will use this CLI. The CLI must be used with a client bundle that authenticates the user. To download the client bundle, authenticate to the MKE UI, then navigate to My Profile, and then click Client Bundles >> New Client Bundle.
- It is recommended that a load balancer be used in front of the API Server to balance user requests across all manager nodes.
- Testing of this STIG was performed on MKE Version 3.7.
- Mirantis Secure Registry (MSR) is designed to serve as a secure and scalable repository for storing, managing, and distributing container images. It includes features and capabilities that enhance the security, reliability, and efficiency of container image management within a Kubernetes environment. It is recommended to use for vulnerability scanning and for managing containers. The scope of this STIG includes critical checks for MSR, but MKE does not require MSR.

### 3. CONCEPTS AND TERMINOLOGY CONVENTIONS

MKE (formerly Universal Control Plane [UCP]) is the industry-leading container orchestration platform for developing and running modern applications at scale.

- **Kubernetes Cluster:** MKE facilitates the deployment and management of Kubernetes clusters. A Kubernetes cluster consists of a set of nodes (physical or virtual machines) that collectively run containerized applications.
- **Node:** Nodes are individual machines within the Kubernetes cluster. MKE manages and orchestrates the deployment of containers across these nodes.
- **Manager Node:** The manager node manages a swarm and persists the swarm state. MKE includes features for managing and maintaining the master node.
- **Worker Node:** Worker nodes host the containers running the applications. MKE distributes containers across worker nodes and ensures their proper operation.
- **Kubernetes API Server:** The API server is a key component of the Kubernetes control plane. MKE interacts with the API server to manage and control the cluster.
- **etcd:** etcd is a distributed key-value store used by Kubernetes to store configuration data. MKE includes tools for managing and maintaining the etcd data store.
- **Cluster Management:** MKE provides features for easy cluster creation, scaling, and management. This includes adding or removing nodes from the cluster, upgrading the Kubernetes version, and ensuring high availability.
- **RBAC (Role-Based Access Control):** MKE supports RBAC, allowing administrators to define roles and permissions for users and groups within the cluster. This enhances security and access control.
- **Integration with Docker:** Mirantis MKE is often tightly integrated with Docker technologies, leveraging Docker for container runtime and orchestration.
- **Security Features:** MKE includes security features such as image signing, secure communication, and support for security policies to ensure a secure containerized environment.
- **Swarm:** Previously Docker Swarm, an orchestration engine. MKE affords users the ability to deploy containers managed with either Docker Swarm or Kubernetes as the orchestration engine.
- Some Kubernetes components are run as Swarm services because the MKE control plane is itself a Docker Swarm cluster.

#### 3.1 MKE Architecture

MKE is a centralized place with a graphical UI to manage and monitor Kubernetes and/or Swarm cluster instance. The core MKE components are:

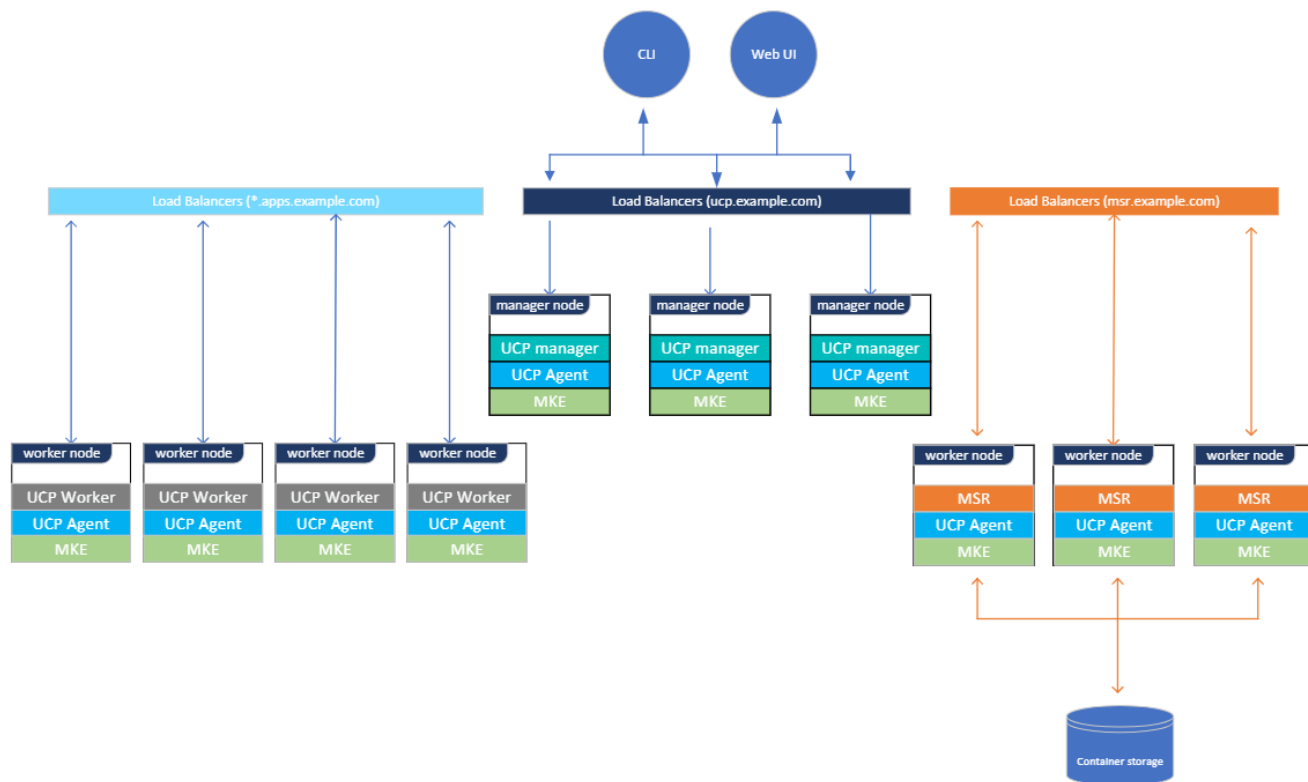
- **ucp-cluster-agent:** Reconciles the cluster-wide state, including Kubernetes add-ons such as Kubecompose and KubeDNS, managing replication configurations for the etcd and RethinkDB clusters, and syncing the node inventories of SwarmKit and Swarm Classic. This component is a single-replica service that runs on any manager node in the cluster.

- **ucp-manager-agent:** Automatically deploys all MKE components on manager nodes including the MKE web UI and the data store that MKE uses. Reconciles the node-local state on manager nodes, including the configuration of the local Docker daemon, local data volumes, certificates, and local container components. Each manager node in the cluster runs a task from this service.
- **ucp-worker-agent:** Performs the same reconciliation operations as ucp-manager-agent, but on worker nodes. This component runs a task on each worker node.

Users interact with MKE either through the web UI or the CLI. With the MKE web UI, users can manage their swarm, grant and revoke user permissions, deploy, configure, manage, and monitor applications.

MKE exposes the standard Docker API, so users can continue using such existing tools as the Docker CLI client. As MKE secures the cluster with RBAC, users must configure Docker CLI client and other client tools to authenticate requests using client certificates, which are available for download from the MKE profile page.

**Figure 3-1: MKE Architecture**



## 4. GENERAL SECURITY REQUIREMENTS

### 4.1 Comprehensive Security

- Access Control and Authentication:
  - Implement RBAC to define and enforce user roles and permissions within the MKE cluster.
  - Integrate with identity providers such as LDAP or Active Directory for centralized user authentication.
- Secure Communication:
  - Enforce the use of TLS/SSL for securing communication channels within the MKE cluster.
  - Use certificates issued by trusted Certificate Authorities (CAs) to authenticate MKE components.
- Kubernetes RBAC:
  - Leverage Kubernetes RBAC features to restrict access to sensitive resources and API endpoints.
- Image Security:
  - Employ signed container images to ensure the integrity and authenticity of images.
- Network Security:
  - Utilize network policies to control the flow of traffic between pods and enforce network segmentation.
- Audit Logging:
  - Enable and configure audit logging to capture relevant events within the MKE cluster.
  - Regularly review and analyze audit logs for security incidents and policy violations.
- Update and Patch Management:
  - Ensure the MKE platform and underlying infrastructure is updated with the latest security patches.
  - Regularly monitor for updates and apply patches promptly to address known vulnerabilities.
- Secure etcd Configuration:
  - Implement security measures for the etcd datastore, including encryption of data in transit and at rest. Restrict access to the etcd cluster to only necessary components and nodes.