DOCKER ENTERPRISE 2.x LINUX/UNIX SECURITY TECHNICAL IMPLEMENTATION GUIDE (STIG) OVERVIEW

Version 1, Release 1

19 July 2019

Developed by Docker and DISA for the DoD

Trademark Information

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

1. INTRODUCTION

1.1 Executive Summary

This Docker Enterprise 2.x Linux/UNIX Security Technical Implementation Guide (STIG) provides the technical security policies, requirements, and implementation details for applying security concepts to container platforms that are built using the Docker Enterprise product suite, specifically for Linux and UNIX, which is built and maintained by Docker, Inc. It also incorporates high-level technical guidance for establishing a secure software supply chain, using the Docker Enterprise platform in concert with containers based on a standard image format and runtime prevalent in industry. The Docker platform is designed to give both developers and IT professionals the freedom to build, manage, and secure mission-critical applications with no technology or infrastructure lock-in. Docker enables a secure, trusted software supply chain workflow which is used as the foundation for building software and for deploying applications onto any infrastructure; from traditional on-premises datacenters, to public cloud providers.

1.2 Authority

DoD Instruction (DoDI) 8500.01 requires that "all IT that receives, processes, stores, displays, or transmits DoD information will be […] configured […] consistent with applicable DoD cybersecurity policies, standards, and architectures" and tasks that Defense Information Systems Agency (DISA) "develops and maintains control correlation identifiers (CCIs), security requirements guides (SRGs), security technical implementation guides (STIGs), and mobile code risk categories and usage guides that implement and are consistent with DoD cybersecurity policies, standards, architectures, security controls, and validation procedures, with the support of the NSA/CSS, using input from stakeholders, and using automation whenever possible." This document is provided under the authority of DoDI 8500.01.

Although the use of the principles and guidelines in these SRGs/STIGs provides an environment that contributes to the security requirements of DoD systems, applicable NIST SP 800-53 cybersecurity controls need to be applied to all systems and architectures based on the Committee on National Security Systems (CNSS) Instruction (CNSSI) 1253.

1.3 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

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

1.4 STIG Distribution

Parties within the DoD and Federal Government's computing environments can obtain the applicable STIG from the Cyber Exchange website. This site contains the latest copies of any STIGs, SRGs, and other related security information. The address for the Cyber Exchange site is https://cyber.mil/. Those without a CAC that has DoD Certificates can use site https://public.cyber.mil/.

1.5 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 For Official Use Only (FOUO) will be available for those 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.

1.6 Document Revisions

Comments or proposed revisions to this document should be sent via email to the following address: 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.

1.7 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 Authorizing Official. 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 Department of Defense. While other agencies and organizations are free to use it, care must be given to ensure that all applicable security guidance is applied both at the device hardening level as well as the architectural level due to the fact that some of the settings may not be able to be configured in environments outside the DoD architecture.

1.8 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). DoD Authorizing Officials (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. This documentation is not published for general access to protect the vendor's proprietary information.

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

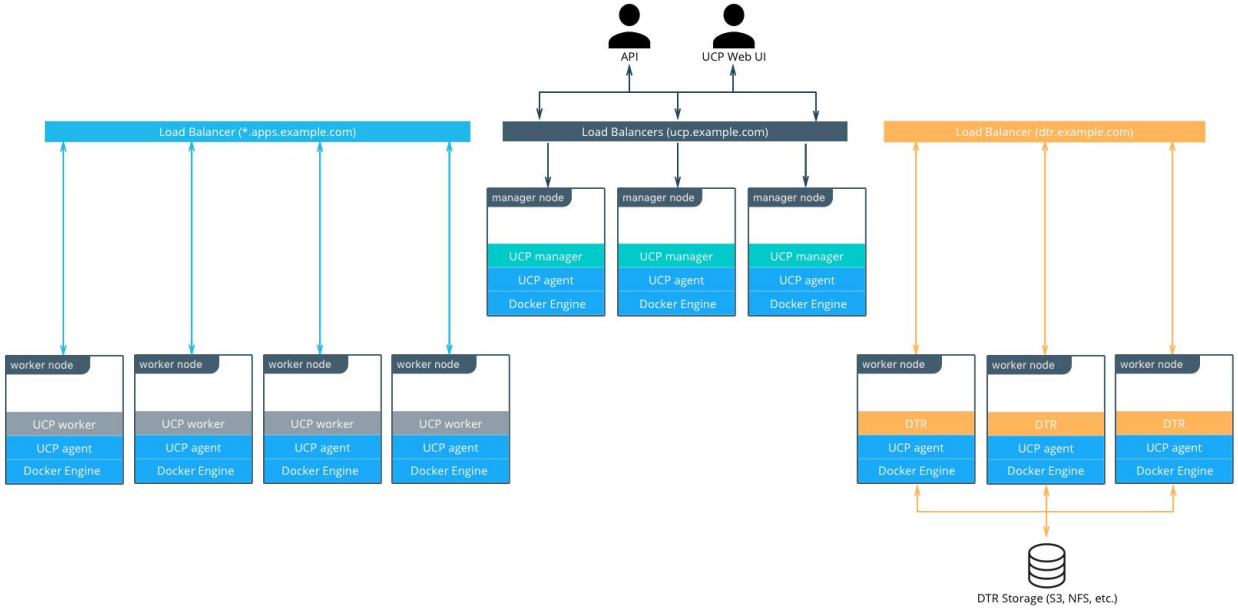
- National Information Assurance Partnership (NIAP) evaluation for National Security Systems (NSS) (http://www.niap-ccevs.org/) IAW CNSSP #11
- National Institute of Standards and Technology (NIST) Cryptographic Module Validation Program (CMVP) (http://csrc.nist.gov/groups/STM/cmvp/) IAW Federal/DoD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

2. ASSESSMENT CONSIDERATIONS

2.1 Security Assessment Information

A standard, production-level architecture for Docker Enterprise uses 10 nodes: 3 UCP controllers, 3 nodes for the DTR, and 4 worker nodes for application workloads. Typical node sizing starts at 4 core and 16gb of RAM. The number of worker nodes is arbitrary, and most environments will have more depending on application needs; it doesn't change the architecture or the cluster configuration. Access to the environment is done through 3 load balancers (or 3 load balancer virtual hosts) with corresponding DNS entries for the UCP controllers, the DTR replicas, and the applications running in the cluster. The DTR replicas use shared storage for images. S3-compatible object storage or NFS storage.

Figure 2-1: Sample Production Deployment



3. CONCEPTS AND TERMINOLOGY CONVENTIONS

3.1 Container Runtime and Image Standards

In June 2015, Docker, Inc. and other industry vendors developed standard specifications on which all containers and their respective platforms can be based. This set of standards, known as the Open Container Initiative (OCI) has been adopted and implemented by Docker, Inc. and a number of other vendors.

OCI includes two primary specifications: the Runtime Specification (runtime-spec) and the Image Specification (image-spec). The Runtime Specification defines how all containers are unpacked and run on an underlying operating system. The open source runC project is an example of the OCI Runtime Spec and is one of the components on which the Docker Enterprise platform is built. The Image Specification defines how software can be packed as a bundle for subsequent unpacking as a container as defined by the Runtime Specification. The Dockerfile format and the builder implementation provided by the Docker Engine are examples of the OCI Image Spec in practice.

It is important to note that this STIG does not provide an exhaustive set of secure technical recommendations on the use and hardening of OCI-compliant containers and images. However, since containers and their respective images represent the core value proposition on the use of a container platform such as Docker Enterprise, one should reference published standards for securing the containers and images themselves (e.g., NIST SP 800-190) and take into account the potential residual risk posed from their use when using the Docker Enterprise platform.

3.1.1 Container Fundamentals and Concepts

A container is a standard unit of software that packages code and all its dependencies so the application runs quickly and reliably from one computing environment to another. A Docker container image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries, and settings.

Container images become containers at runtime and in the case of Docker, container images become containers when they run on the Docker Engine. Containerized software will always run the same, regardless of the infrastructure. Containers isolate software from its environment and ensure that it works uniformly despite differences, for instance, between development and staging. Container characteristics are as follows:

- **Standard:** Docker follows the OCI standard for containers, so they could be portable anywhere
- **Lightweight:** Containers share the machine's operating system (OS) system kernel, and therefore do not require an OS per application
- **Secure:** Applications are safer in containers when they are isolated from each other and the OS

3.1.2 Docker Enterprise Overview

Docker Enterprise is the umbrella container platform product on which this STIG is based. It is made up of the following core components:

- Docker Engine - Enterprise
- Universal Control Plane (UCP)
- Docker Trusted Registry (DTR)

Docker Engine - Enterprise is the core container engine and runtime that which is installed on a supported Linux host operating system. This is also referred to as a node. Multiple nodes of Docker Engine - Enterprise running on separate hosts can be clustered together using a built-in clustering mechanism known as Swarm Mode, which is provided by the Engine. On top of this cluster sits the UCP container orchestration and management platform, which itself runs as a set of containers. DTR also runs as a set of containers, but is managed by UCP.

Throughout this document and the STIG, the term “Docker Enterprise cluster” is used, which refers to a cluster of Docker Engine - Enterprise nodes, on which both UCP and DTR are running.

4. GENERAL SECURITY REQUIREMENTS

4.1 Docker Enterprise Security Requirements

Docker Enterprise provides an end-to-end overlay of security functionality, all built on the foundational Docker Engine container runtime. It is important to note that no one component of the Docker Enterprise stack as described below, including the Engine, can be used by itself to meet the security recommendations set forth in the STIG. They must all be used in concert with one another to meet the breadth of required security controls.

4.1.1 Docker Engine - Enterprise

At the bottom of the stack lies the Docker Engine - Enterprise container runtime. This is the core component on which all other components of Docker Enterprise and general containerized workloads and applications reside. The Engine provides critical security functions for the entire platform, namely FIPS 140-2 validated cryptography and secure Engine clustering via a mutual TLS 1.2 (in AES GCM mode) implementation. These functions provide encryption at rest and encryption in-transit for all metadata and state information for a Docker Enterprise cluster. It is important to note, however, that Docker, Inc's FIPS 140-2 validated cryptography is only embedded into the Docker Engine - Enterprise runtime and has not yet been made available in such a way as to protect the remainder of components that make up Docker Enterprise.

4.1.2 Universal Control Plane (UCP)

On top of Docker Engine - Enterprise nodes sits a set of containerized system services known as the Universal Control Plane. This is the primary management component via which all container scheduling, resource allocation, and access control policies can be implemented. The majority of security recommendations outlined in this STIG are configured within UCP as it provides the core integrations with directory services (e.g., LDAP), access control enforcement, and content integrity checking and enforcement for an entire Docker Enterprise cluster.

4.1.3 Docker Swarm

Docker Swarm is the default container orchestrator

4.1.4 Docker Trusted Registry (DTR)

Docker Trusted Registry is a containerized server side application that stores and lets you distribute Docker images. DTR is used to control where your images are stored and to integrate storage and distribution in an in-house development workflow.

4.1.5 Kubernetes

Included as part of UCP 3.1 is a distribution of Kubernetes v1.11, which is an open source container orchestrator. All of the core functionality provided by the upstream Kubernetes project is made available within UCP. This includes all object types such as pods, deployments, jobs,

services, and load balancers to name a few. In addition, the FIPS 140-2 validated cryptography provided by the underlying Docker Engine - Enterprise runtime supports some of the security functions built in to Kubernetes, more specifically Kubernetes secrets and the cluster metadata.

Two things are important to note. First, Docker, Inc's FIPS 140-2 validated cryptography is only embedded into the Docker Engine - Enterprise runtime and has not yet been made available in such a way as to protect the remainder of components that make up Docker Enterprise, including Kubernetes. Secondly, the Kubernetes features and functions are not supported by Docker, and as such, Kubernetes requirements are not included in the STIG.