CONTAINER PLATFORM
SECURITY REQUIREMENTS GUIDE (SRG)
TECHNOLOGY OVERVIEW

Version 1, Release 4

26 July 2023

Developed by DISA for the DOD

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

LIST OF TABLES

Page

1. INTRODUCTION

1.1 Executive Summary

The Container Platform Security Requirements Guide (SRG) will provide technical requirements for securing a container platform. For this document, a container platform will be defined as high-level software with the capability of managing container lifecycles. A container platform is composed of a container engine or runtime, container registry, and key-value store (i.e., components). With the platform, services such as a DNS, firewall, router, and web console may also be deployed. These services must follow the appropriate Security Technical Implementation Guides (STIGs) and Security Requirements Guides (SRGs) for the technology, as well as any guidance for the manner in which these services are implemented.

1.1.1 Security Requirements Guides

SRGs are collections of requirements applicable to a given technology family. SRGs represent an intermediate step between Control Correlation Identifiers (CCIs) and STIGs. CCIs represent discrete, measurable, and actionable items sourced from Information Assurance (IA) controls defined in a policy, such as the National Institute of Standards and Technology (NIST) Special Publication (SP) 800-53. STIGs provide product-specific information for validating and attaining compliance with requirements defined in the SRG for that product's technology area.

There are four core SRGs: Application, Network, Operating System, and Policy. Each addresses the applicable CCIs in the context of the technology family. Subordinate to the core SRGs, Technology SRGs are developed to address the technologies at a more granular level.

This Container Platform SRG is based on the Application SRG. It contains general check and fix information that can be used for products for which STIGs do not exist.

The STIGs based on this SRG will provide the product-specific technical implementation guidance for that product. The STIG will contain the specific check and fix information for the product it covers.

SRG Hierarchy example:

Application SRG
|__Database SRG
* |__MS SQL Server 2005 STIG*

The SRG relationship and structure provides the ability to identify requirements that may be considered not applicable for a given technology family and provide appropriate justification. It also provides the structure to identify variations in specific values based on the technology family. These variations will be captured once and will propagate down to the Technology SRGs and then to the STIGs. This will eliminate the need for each product-specific STIG to address items that are not applicable.

1.1.2   SRG Naming Standards

In an effort to establish consistency across the SRGs, a naming standard for the Group Title and STIGIDs has been established.

Technology SRG Naming Standards

For Technology SRG Group Title and STIGIDs the following applies:

{Core SRG value}-{Technology SRG}-{5- or 6-digit numeric sequence number}

Examples:

> *SRG-NET-000001-RTR-000001*
> *SRG-APP-000001-COL-000001*
> *SRG-NET-000001-VVSM-00001*
> *SRG-OS-000001-UNIX-000001*

Checks/fixes will be included at this level in a general form. These checks and fixes will apply for any STIGs that are created for products that do not have product-specific check and fix guidance.

1.2   Authority

Department of Defense Instruction (DODI) 8500.01 requires that "all IT [information technology] that receives, processes, stores, displays, or transmits DOD information will be […] configured […] consistent with applicable DOD cybersecurity policies, standards, and architectures." The instruction tasks that 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 [National Security Agency/Central Security Service], 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 must be applied to all systems and architectures based on the Committee on National Security Systems (CNSS) Instruction (CNSSI) 1253.

1.2.1   Relationship to STIGs

SRGs define the requirements for various technology families, while STIGs provide the technical implementation guidelines for specific products. A single SRG/STIG is not all-inclusive for a given system, which may include, but is not limited to Database, Web Server, and

Domain Name System (DNS) SRGs/STIGs. For a given system, compliance with all (multiple) SRGs/STIGs applicable to a system is required.

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   SRG and 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/.

1.5   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.

1.6   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. 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 CONSIDERATIONS

2.1 NIST SP 800-53 Requirements

All applicable baseline technical NIST SP 800-53 requirements and security best practice requirements are included in this SRG.

CNSSI 1253 defines the required controls for DOD systems based on confidentiality, integrity, and availability (baseline) of the given information system. In all cases, CNSSI 1253, along with required baselines, will serve as the policy requirement for any given asset or information system.

2.2 General Procedures

This SRG has procedures that are intended to provide appropriate evaluation and remediation functions for a typically configured system. These procedures are not product specific and are intended for use when a product-specific STIG is not available.

3. CONCEPTS AND TERMINOLOGY CONVENTIONS

A container platform enables DevSecOps personnel, or automation working on their behalf, to pull images from container platform registries, deploy those images into containers, and manage the running containers. This deployment process results in a usable version of the application that is running and ready to respond to requests. When an image is instantiated into a container, the image itself is not changed; instead, a copy of it is placed within the container platform and transitioned from being a dormant set of application code to a running instance of the application.

The abstraction provided by a container platform allows DevSecOps personnel to specify how many containers of a given image need to be running and what resources, such as memory, processing, and disk, need to be allocated to each. The container platform knows the state of each host within the cluster, including what resources are available for each host, and determines which containers will run on which hosts. The container platform runtime then pulls the required images from the registry and runs them as containers with the designated resources.

Container platform tools are also responsible for monitoring container resource consumption, job execution, and machine health across hosts. Depending on its configuration, the container platform runtime may automatically restart containers on new hosts if the hosts they were initially running on failed.

When applications in containers need to be updated, the existing containers are not modified but are destroyed, and new containers are created from updated images. This is a key operational difference with containers: the baseline software from the initial deployment should not change over time, and updates are done by replacing the entire image at once. This approach has significant potential security benefits because it enables organizations to build, test, validate, and deploy exactly the same software in exactly the same configuration in each phase. As updates are made to applications, organizations can ensure that the most recent versions are used.

The container platform runtime should be configured to pull the most up-to-date version of an image from the registry so that the application is always up to date. This "continuous delivery" automation enables developers to build a new version of the image for their application, test the image, push it to the container platform registry, and then rely on the automation tools to deploy it to the target environment. This means that all vulnerability management, including patches and configuration settings, is typically taken care of by the developer when building a new image version.

The container platform keystore provides a reliable way to store sensitive data that needs to be accessed by a system or cluster data. Container platform applications can read from, and write data to, the keystore. This permits applications to reconfigure independently when they change.

4. GENERAL SECURITY REQUIREMENTS

Container platform security goes beyond configuration settings. To secure a container platform properly, thought needs to be given to the services being hosted, who the user community is, what type of data is being accessed, and where the container platform will reside. By not looking beyond the container platform itself, security flaws in the implementation can lead to the compromise of user personally identifiable information (PII) and organization-sensitive data and processes. This can also compromise access to other systems and applications within the organization with a trusted relationship to container platform services.

4.1 Hosting Operating Systems

The operating system is the foundation for the container platform and any nodes included in the container platform configuration. By not securing the operating system properly, the container platform can become a front end for a nefarious user to gain access to an organization's networked resources.

When securing the container platform, care should be taken to secure files and set user roles and privileges to the least needed for proper operating system and container platform operation. The Container Platform SRG does address operating system file permissions for container platform files, but within an operating system, there are settings and processes that are outside the realm of the Container Platform SRG. There may also be instances where a container platform requirement can be met, but without the operating system requirement being met, the container platform is not fully secure. To secure the operating system properly, the appropriate Operating System SRG or specific vendor STIG should be used.

4.1.1 Roles

Properly defining user roles is essential to securing the container platform. Too often, all operating system users are given the same roles. Giving users more privileges than necessary allows a user who is not part of the container platform administrator role privileges to make container platform changes. Looking at the roles that the organization wants to implement for privileged users and giving users only the roles required for carrying out each user's duties is crucial. The definition and duties of each role should be provided before any user accounts are created and before the container platform is deployed.

4.1.2 File Permissions

When securing an operating system, many of the requirements rely on privileges and ownership of files. The permissions laid forth by the operating system requirements may or may not be stricter than permissions of the container platform requirements for privileges and ownership. Care must be taken to give the least privileges and ownership to container platform files and still allow operation of the container platform and the hosted services. To arrive at the proper least privilege settings for the container platform and hosted services, a test environment must be used

along with a well-developed test plan to ensure the production container platform operates properly and as expected.

4.2   Container Platform Management

Container platform management is the process of providing administrative duties in the configuration, deployment, and sustainment of the container platform software, components, and user services. The management duties can be performed through local (i.e., console) or remote access.

Remote access can take many forms, such as through the internet or a dedicated management network.

When the management is done locally, the hosting hardware and operating system perform the validation of users, assign permissions or privileges to the user, and enforce file protections. The major security concern during local management of a container platform is constraining the user to only those files and functions needed to perform their duties.

Remote access has the added security concern of the transmission of data. All remote management to a container platform must be encrypted. The encryption of the traffic should begin at the start of the transmission session. The loss of administrative credentials during a non-encrypted session would negate any security that encryption of later traffic would add. Several methods of performing administrative activities remotely are through third-party software that is used specifically to administer the container platform (e.g., web console), through secure shells and virtual private networks (VPNs), or through a dedicated management network.

Remote access must also be controlled and not easily available and viewable by non-administrative users. Where local access can be controlled through physical barriers, remote access needs to be controlled through electronic barriers such as access lists or management networks. Care should be taken when implementing remote access technologies not to bypass security measures already in place to protect the container platform.

No matter the method used for container platform management, users must be authenticated using DOD-approved PKI credentials. The validation of DOD-approved PKI credentials will not be done by the container platform itself but will be performed by the operating system or local and remote access host system. If a web console is provided, the web service must authenticate the users via DOD-approved PKI authentication before granting access to the web console.

4.3   Container Platform Component Authentication

Trusted and secure communication between container platform components is essential. In addition to having a secure connection, the relationship must be trusted. For a component to identify itself, X.509 certificates are used. These certificates must be protected and tied to a trusted certificate authority (CA). By using only certificates from a trusted CA, the use of self-signed certificates is not permitted. Within a DOD environment, the system should be configured to work in accordance with the DOD PKI/PKE policy.

4.4   Transmitted Data Protection

Transmitted data must be protected, whether between the user and a service or the container platform, inter-component, or data image pulls. If an adversary were able to compromise the data, the entire container platform would be compromised. A common method of securing communications is the use of a protocol that provides data integrity and encryption services. Transport Layer Security (TLS) is more effective than SSL for improving privacy and data security for communications between applications. The information systems must use TLS v1.2 at a minimum to secure the container platform architecture.

4.5   Conclusion

The container platform is an ecosystem within itself, made up of the host systems, container platform components, such as DNS servers, firewalls and routers, and user services. Securing the overall container platform must take into consideration each one of these parts, the communication between the parts, the user community, and the data being processed. This SRG does not address every component or service offered or needed by the overall container platform because there is already security documentation within SRGs, STIGs, and guides that addresses them. Some questions that can be asked to help further secure the container platform are:

- Are the user container images following application guides and container best practices?
- Are all the components and services hardened according to guidance within the technology SRG or more specific technology STIG?
- Are all components and services following ports and protocol guidelines set forth by DOD Instruction 8551.01 policy?
- Are all endpoints trusted and communication channels encrypted?
- Are workloads, user groups, and data sensitivity levels being isolated properly?
- Is there a process for introducing new container images into the production environment?
- Are the security tools monitoring the components and user services container aware and designed to operate at the scale and change rate typically seen with containers?
- Are the containers run with policies to limit resource usage such as CPU, storage, and memory?

To fully secure the container platform, the system administrators must fully understand how the container platform is installed and the services and data the container platform will host. This goes beyond the container platform SRG, but it must be performed to guarantee an overall secure system.