Container Guidance for Federal Information Systems

Trevor Vaughan
Onyx Point, Inc.

Version 1.0
May 21, 2019

This Work was created in the performance of a Cooperative Research and Development Agreement (CRADA Number TD-0002-16) with the National Security Agency, The Government of the United States has certain rights to use this Work.

1 Introduction

This document is targeted at helping users, administrators, and security officers understand where the container ecosystem aligns with current published guidance for federal information systems[1]. It is important to note that this document is not intended to provide instruction on specifically how to run a container-based infrastructure. It also does not pertain to securing the operating system upon which the containers are running. There is ample guidance available for performing risk-based configuration of general purpose operating systems[2,3,4,5,6,7].

This document is meant to be prescriptive guidance. The words SHALL, MAY, MUST, MUST NOT, SHOULD, and SHOULD NOT should be interpreted in as defined by RFC 2119.

This document is written in such a way that it may be incorporated directly into organizational policy. Any and all guidance from official sources overrides anything in this document.

This document is not official federal guidance

Federal organizations should use the guidelines presented in the NIST 800-18[8] Guide for Developing Security Plans for Federal Information Systems as the foundation for container-focused System Security Plans (SSP).

Please address corrections or comments to info@onyxpoint.com with the text ***[CONTAINER-GUIDANCE]*** in the subject of the message.

[1] 40 U.S.C., Sec. 11331 - https://www.govinfo.gov/app/details/USCODE-2011-title44
[2] NIST 800-53 - https://nvd.nist.gov/800-53
[3] NIST 800-171 - https://csrc.nist.gov/publications/detail/sp/800-171/rev-1/final
[4] CNSSI 1253 - https://www.cnss.gov/CNSS/issuances/Instructions.cfm
[5] DISA STIG - https://iase.disa.mil/stigs/os/Pages/index.aspx
[6] NIST 800-123 - https://csrc.nist.gov/publications/detail/sp/800-123/final
[7] Center for Internet Security - https://www.cisecurity.org/cis-benchmarks/
[8] NIST 800-18 https://csrc.nist.gov/publications/detail/sp/800-18/rev-1/final

2 Terminology

This document uses several terms that may be confusing based on general usage. To clarify intent, these terms are defined below.

2.1 Host Operating System

The host operating system is the operating system software upon which the container infrastructure runs. Common examples include Linux, Microsoft Windows, and Mac OS.

2.2 Container

Container permissions are confined by the host operating system. The relationship between containers and the host operating system is comprehensively covered in NIST 800-190[9]. For the purposes of this document, a container should be specifically thought of as:

- A filesystem housing executable files and configuration files
- A set of running processes based upon items in the container filesystem

2.3 Container Image

A container image is a reusable set of software and related configurations that, when added to a container infrastructure, results in a running container. Common examples include Open Container Initiative (OCI)[10] and Docker[11] images.

2.4 Container Infrastructure

A container infrastructure is the software that manages the execution of containers on top of the host operating system. Guidance for container infrastructures and the surrounding workflow is well covered in NIST 800-190.

Container infrastructures may include, but not be limited to, the following types:

- Single host container management. Examples include systemd[12], RKT,[13] and Docker
- Multi-host or clustered container management . Examples include Kubernetes[14], and Docker Swarm.

[9] NIST 800-190 §2.2 - https://csrc.nist.gov/publications/detail/sp/800-190/final
[10] Open Container Initiative - https://www.opencontainers.org/
[11] Docker - https://www.docker.com/
[12] systemd - https://www.freedesktop.org/wiki/Software/systemd/
[13] RKT - https://coreos.com/rkt/
[14] Kubernetes - https://kubernetes.io/

3 System Security Plans for Containers and Container Infrastructures

Given that containers are simply a set of installed applications running within a confined namespace on the host operating system, they should generally be categorized as Major or Minor Applications. In this case, they will clearly inherit many security controls from the host operating system and container infrastructure.

OMB Circular A-130, Appendix III defines a **major application** as one “that requires special attention to security due to the risk and magnitude of harm resulting from the loss, misuse, or unauthorized access to or modification of the information in the application”.

NIST SP 800-18 defines a **minor application** as one, “other than a major application, that requires attention to security due to the risk and magnitude of harm resulting from the loss, misuse, or unauthorized access to or modification of information in the application.” It also notes that minor applications do not generally require their own system security plan since their security controls are handled by the larger major application or general support system within which they are contained.

For the purposes of this document, we are recommending that all containers that are not identified as major applications be treated as minor applications without exception. This ensures that all necessary documentation can be provided along with the associated container image and reduce the overall time to field for all users.

The container infrastructure shall be categorized as a ***Major Application*** since it plays a critical role in securing access to, and management of, all running containers. If the container infrastructure is directly integrated into the underlying host operating system, it may be covered under existing operating system security guidance. However, for this premise to be valid, the configuration guidance must specifically address the container management capabilities of the subsystem.

4 General Guidance

The following is provided as general guidance in the creation of a suitable SSP for containers and container environments. The container landscape is changing rapidly, so this is intended to focus on practical capabilities available via current general vendor supported capabilities. We acknowledge that new technologies and techniques will make the container landscape far more secure than it is presently but the level of turmoil in the current ecosystem prevents any level of prescriptive guidance at this time.

If at all possible, the relevant portions of the NIST 800-53 and/or CNSSI 1253 controls will be associated with the provided guidance to provide ease of integration into a full system SSP. Items that have been concretely derived will be noted as mandatory while items that pertain to general best practice will be noted as optional.

4.1 Building Containers

1. Packages and software installed inside of a container SHALL be part of an approved software baseline[15] and MUST follow all requirements for normal software installation[16].
    a. Containers are simply collections of running software[17] and associated configurations. As such, they are subject to all of the same rules and regulations as any other installed and running software.

2. All container images MUST be cryptographically signed[18]
    a. Container images are simply a packaging format. Like all other software, it must be cryptographically validated prior to installation onto a federal information system.

3. Containers MUST include the fewest number of running services to meet functional requirements[19].
    a. This does **not** mean that there must strictly be one application per container as is often recommended. Organizations will need to evaluate the maintenance burden of their container infrastructure components and may discover that different services require different included capabilities.
    b. There are legitimate functional reasons for many unorthodox container composition practices and therefore an inflexible approach should not be mandated organizationally.

4. Per NIST 800-53 and CNSSI 1253[20], all cryptography used for the protection of sensitive information within a given container, or a container infrastructure, on a federal information system MUST use a NIST-validated, or NSA-approved, cryptographic engine [21].
    a. Personnel designing the container infrastructure and individual containers are responsible for documenting all cryptographic endpoints, their use, and the FIPS 140-2 status of the utilized software in the container SSP.

[15] NIST 800-53 - CM-2. CNSSI 1253 - I:LMH
[16] NIST 800-53 - CM-11. CNSSI 1253 - C:LMH, I:LMH
[17] NIST 800-53 - CM-8, CNSSI 1253 - I:LMH
[18] NIST 800-53 - CM-7(5), CNSSI 1253 - C:H, I:H
[19] NIST 800-53 - CM-7, CNSSI 1253 - C:LMH, I:LMH
[20] NIST 800-53 - SC-13, CNSSI 1253 - C:LMH, I:LMH
[21] NIST CMVP - https://csrc.nist.gov/projects/cryptographic-module-validation-program/standards

   b. It is functionally possible to defer the cryptographic capabilities to other containers or the host operating system. This must be explicitly and clearly documented as an item deferred to an external component in the container SSP for acceptance and resolution by the container infrastructure operators.
      i. Authorizing officials should note that this approach requires the traffic on the underlying host operating system to be passed without cryptographic protection. The underlying host operating system has purview over all running containers, so protection against the underlying host is impractical.

5. The ability to build all running container images MUST be validated on a regular basis[22].
   a. The inability to rebuild a container presents a distinct operational risk since the discovery of a vulnerability in that image would preclude rapid rebuilding and redeployment of containers based on that image[23].
   b. The inability to rebuild an image should be considered a security finding related to integrity[24].

4.2 Running Containers

1. The host operating system SHALL be secured in accordance with an approved standard [25].
   a. The fact that a host is running a container infrastructure does not preclude it from configuration in accordance with applicable Federal policy.

2. Container image cryptographic signatures MUST be validated prior to execution[26].
   a. There are many methods for doing this from simple detached signatures to integrated systems that work with the underlying container infrastructure. At this time, there is no reason to not provide this capability as part of a manual workflow or scripted automation pipeline[27].

3. Containers MUST NOT allow remote shell access unless operationally required[28].
   a. Organizations may wish to continue using their traditional management tools, which often expect shell access, until they can migrate to a more cloud-native approach.
   b. This correlates with the requirement that systems run no unnecessary services and, inversely, also means that the host operating system may not need to run

[22] NIST 800-53 - SA-12, CNSSI 1223 - C:H, I:H, A:H
[23] NIST 800-53 - CP-2, CNSSI 1253 - A:LMH
[24] NIST 800-53 - SI-2, CNSSI 1253 - I:LMH
[25] FISMA - https://csrc.nist.gov/topics/laws-and-regulations/laws/fisma
[26] NIST 800-53 - CM-7(5), CNSSI 1253 - C:H, I:H
[27] NIST 800-53 - SI-3, CNSSI 1253 - I:LMH
[28] NIST 800-53 - AC-3, CNSSI 1253 - I:LMH

remote shell services on a regular basis[29].

4. All containers MUST run in a strictly isolated namespace, backed by mandatory access control (MAC) capabilities with the minimum number of required system capabilities[30].
    a. Different host operating systems have different methods for providing container isolation.
    b. Any host operating system that cannot provide for restricted system capabilities or MAC isolation must not be used to run a container infrastructure handling operational data.
    c. The system capabilities and network access required by a container must be documented as part of the component SSP and provided for approval to the operator of the target container infrastructure.
    d. Management containers may be used as an entry point to manage the underlying host operating system but must be documented as part of the host operating system SSP and explicitly approved for use by the cognizant security officer. It must be explicitly clear when a container has elevated host privileges since these containers provide entry points to accessing other containers on the system and/or host network traffic. These containers must be monitored as high-risk to the underlying host operating system.

5. Containers SHOULD NOT be run as a privileged user[31].
    a. Modern container execution environments have gained the ability to run containers as non-privileged users and should be utilized to ensure that the container infrastructure itself cannot be compromised from within a running container.
    b. If your system has the ability to run as a non-privileged user, then this section applies as MUST NOT instead of SHOULD NOT.
    c. Once this capability becomes mainstream, this will change to MUST NOT in all cases.

6. Containers MUST be run in ***read-only*** mode when processing sensitive data or as a critical component of an operational workflow[32].
    a. This does not apply to containers in development as long as they do not process sensitive data.
    b. This does not apply to operational containers as long as they cannot do the following:
        i. Access sensitive data
        ii. Disrupt critical workflows

[29] NIST 800-53 - CM-7, CNSSI 1253 - C:LMH, I:LMH
[30] NIST 800-53 - CM-7, CNSSI 1253 - C:LMH, I:LMH
[31] NIST 800-53 - AC-6, CNSSI 1253 - C:MH, I:MH
[32] NIST 800-53 - CM-7, CNSSI 1253 - C:LMH, I:LMH

   c. Running a container in non-read-only mode is a security relevant event and must be audited as such.
   d. Note: If an alternate method of configuration monitoring and correction is used as part of an overall management solution then a running container may be considered a management endpoint and treated as any other managed component. In this case, the read-only requirement becomes **optional** for containers that are registered into the existing management solution.

7. Container images MUST NOT be transferred from an environment of greater sensitivity to one of lower sensitivity unless explicitly authorized[33].
   a. There is no foolproof method for ensuring that inappropriate information from a high sensitivity environment has been purged from a container image. The base artifacts should be passed to the environment of lower sensitivity and the container image rebuilt at that level.
   b. This has the added benefit that each environment is then self-sustaining and does not require regular transferral of material from a higher sensitivity environment to one of lower sensitivity.
   c. In the case of transferral between environments of differing classification, this becomes **mandatory**[34].

8. The ability to update cryptographic key material (X.509 keys, etc…) in a timely manner if close to expiration or post-compromise SHALL be documented and have demonstrable validation tests[35].
   a. The inability to properly protect information or persist valid cryptographic connections critically compromises information system capabilities[36].

9. Teams that provide container images SHALL NOT be the same teams that run those containers in production environments[37].
   a. There are two exceptions to this:
      i. Containers that are run for administrative maintenance of the system by the operations team. Administrative containers must still follow all other guidance in this document.
      ii. Containers created for live troubleshooting. These containers may be created if all other methods of troubleshooting a system have failed. Troubleshooting containers MUST NOT remain active after the fix has been applied to the source container image.

[33] NIST 800-53 - SC-4, CNSSI 1253 - C:MH
[34] NIST 800-53 - MP-4, CNSSI 1253 - C:MH, I:MH
[35] NIST 800-53 - SC-12, CNSSI 1253 - C:LMH, I:LMH
[36] NIST 800-53 - SC-12(1), CNSSI 1253 - A:H
[37] NIST 800-53 - AC-5, CNSSI 1253 - C:MH, I:MH

4.3 Auditing Containers

1. Containers MUST be evaluated for common vulnerabilities prior to execution[38].
    a. This correlates closely with the requirement for running containers in read-only mode since running containers are notoriously difficult to audit thoroughly when users are able to modify them at run time.

2. Running containers that deviate from their approved base image MUST be flagged for assessment[39].
    a. Live container modifications is an anomaly in production systems. Data correlation may be used to identify false positives based on running containers that have been approved for modification.

3. Container images MUST be evaluated for common vulnerabilities on a regular, organization-defined, basis[40]. Running containers based on vulnerable images must be remedied within a well defined time limit.
    a. It is recommended that the time limit correlate with updates to traditional host security evaluation schedules for operating systems.

4. Host volumes that are mounted into a container MUST be clearly documented into the SSP to include the nature and purpose of the mount[41].
    a. The ability for a container to access components in the underlying host provides potential for data exfiltration or inappropriate information access.
    b. Any volume mount is an auditable event.

5. The creation or destruction of containers in a container infrastructure MUST be audited, by the container infrastructure[42].
    a. The addition of new services to a system is a security relevant change and should be logged as such.

6. Connection of a container to a network other than one restricted to the underlying host operating system MUST be audited[43].
    a. Network connections require approvals since each running service is a potential access point into your infrastructure.

[38] NIST 800-53 - RA-5, CNSSI 1253 - C:LMH, I:LMH, A:LMH
[39] NIST 800-53 - CM-7, CNSSI 1253 - C:LMH, I:LMH
[40] NIST 800-53 - RA-5, CNSSI 1253 - C:LMH, I:LMH, A:LMH
[41] NIST 800-53 - CM-7, CNSSI 1253 - C:LMH, I:LMH
[42] NIST 800-53 - CM-8, CNSSI 1253 - I:LMH
[43] NIST 800-53 - SC-10, CNSS-1253 - C:MH, I:MH

Future Considerations

The container landscape is rapidly changing and there are many promising technologies that will help further secure both containers and container infrastructures. Future efforts at evaluating the container landscape should assess the adoption of these new technologies and provide practical guidance based on the most stable components available. Particular care should be taken to ensure that any software meets the basic requirements for Federal systems since these are routinely overlooked.

Additionally, hardware-targeted exploits in the same vein as Spectre and Meltdown will continue to evolve and currently available container technologies offer no protection. Organizations should carefully evaluate related risks against multi-tenant systems and the use of combinations of container and virtualization technologies for layered protection.