IS AUDIT/ASSURANCE PROGRAM

APPLICATION CONTAINERS

ISACA®

CONTENTS

ABSTRACT

The *IS Audit/Assurance Program for Application Containers* will assist IT auditors in their assessments of application container deployments.

Audit Subject: Application Containers

Application virtualization allows the number of applications in a hosted environment to be increased without a corresponding increase in the number of servers. Applications can also be 'segmented' into more manageable sizes of data rather than pushing the entire application to a device. A reduced number of servers through virtualization and better deployment techniques help enterprises cut costs and implement changes faster. As enterprises search for ways to implement change even faster, the challenge of maintaining consistency and reliability as software is migrated from one computing environment to another is exacerbated.

Application containers can mitigate this challenge because they consist of the application and all of the application's dependencies, such as libraries and configuration files. To provide assurance that application containers enhance consistency and reliability, ISACA's *IS Audit/Assurance Program for Application Containers* covers preservation of data integrity through all phases of application containerization (planning, development, deployment, maintenance and destruction). Assurance is achieved by tests in the following areas:

- Risk analysis and management
- Security awareness and training
- Images
- Registry
- Orchestrator
- Application security during development
- Secure connections
- Hardening
- Container destruction

Audit Objectives

The primary purpose of this audit program is to assist IT auditors in their assessments of application container deployments. Accordingly, this audit program supports assurance across several domains:

- Clarifying roles and responsibilities, given that developers play a bigger security role in application containerization
- Safeguarding the host operating system by deactivating unnecessary services
- Mitigating risks associated with use of a shared kernel, which is inherent in the application container infrastructure
- Providing confidentiality of network traffic between application containers on the same host

Audit Scope

The audit program addresses the host operating system, network, container runtime and images of application containers, including, but not limited to, Docker® and Rocket®.

Business Impact and Risk

Application containers share the same operating system kernel; if a malicious actor gains access to the container environment, lateral movement is possible and may result in access to all containers. Whenever an enterprise fails to monitor application container environments, unauthorized connections and/or malicious traffic may go undetected. Unauthorized access that remains undetected can result in compromise of data.

Traditional security solutions such as intrusion prevention systems (IPSs) and web application firewalls (WAFs) may be unable to detect vulnerabilities within containers.[1] As a result, adopting a dedicated container security solution is more effective in preventing and detecting threats directed at containers.

Minimum Audit Skills

The audit program assumes that auditors exercise due professional care and possess professional competencies culminating in the proficiency to conduct an audit. ITAF Standard 1005 and ITAC Guideline 2005 (Due Professional Care) require the auditor to exercise professional skepticism and maintain effective communication throughout the course of an audit. ITAF Standard 1006 and ITAF Guideline 2006 (Proficiency) require the auditor to have technical skill, knowledge and/or experience in the areas under assessment. This expectation is particularly relevant to auditors who do not hold the CISA or other appropriate designation.

Testing Steps

Refer to the accompanying spreadsheet file.

[1] US National Institute of Standards and Technology, *Special Publication 800-190 Application Container Security Guide*, September 2017, *https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-190.pdf*

Acknowledgments

ISACA would like to recognize:

Lead Developer

Robin Lyons
CISA, CIA, USA

Expert Reviewers

Larry Marks
CISA, CRISC, CISM, CGEIT, CFE, CISSP, ITIL, PMP, USA

Sergiu Sechel
CISA, CRISC, CISM, CEH, CSSLP, PMP, Romania

Shruti Shrikant Kulkarni
CISA, CRISC, CCSK, CISSP, ITILv3 Expert, UK

ISACA Board of Directors

Rob Clyde, Chair
CISM
Clyde Consulting LLC, USA

Brennan Baybeck, Vice-Chair
CISA, CRISC, CISM, CISSP
Oracle Corporation, USA

Tracey Dedrick
Former Chief Risk Officer with Hudson City Bancorp, USA

Leonard Ong
CISA, CRISC, CISM, CGEIT, COBIT 5 Implementer and Assessor, CFE, CIPM, CIPT, CISSP, CITBCM, CPP, CSSLP, GCFA, GCIA, GCIH, GSNA, ISSMP-ISSAP, PMP
Merck & Co., Inc., Singapore

R.V. Raghu
CISA, CRISC
Versatilist Consulting India Pvt. Ltd., India

Gabriela Reynaga
CISA, CRISC, COBIT 5 Foundation, GRCP
Holistics GRC, Mexico

Gregory Touhill
CISM, CISSP
Cyxtera Federal Group, USA

Ted Wolff
CISA
Vanguard, Inc., USA

Tichaona Zororo
CISA, CRISC, CISM, CGEIT, COBIT 5 Assessor, CIA, CRMA
EGIT | Enterprise Governance of IT, South Africa

Theresa Grafenstine
ISACA Board Chair, 2017-2018
CISA, CRISC, CGEIT, CGAP, CGMA, CIA, CISSP, CPA
Deloitte & Touche LLP, USA

Chris K. Dimitriadis, Ph.D.
ISACA Board Chair, 2015-2017
CISA, CRISC, CISM
INTRALOT, Greece

Robert E Stroud
ISACA Board Chair, 2014-2015
CRISC, CGEIT
XebiaLabs, Inc., USA

Matt Loeb
CGEIT, CAE, FASAE
Chief Executive Officer, ISACA, USA

About ISACA

Nearing its 50th year, ISACA® (isaca.org) is a global association helping individuals and enterprises achieve the positive potential of technology. Technology powers today's world and ISACA equips professionals with the knowledge, credentials, education and community to advance their careers and transform their organizations. ISACA leverages the expertise of its half-million engaged professionals in information and cyber security, governance, assurance, risk and innovation, as well as its enterprise performance subsidiary, CMMI® Institute, to help advance innovation through technology. ISACA has a presence in more than 188 countries, including more than 217 chapters and offices in both the United States and China.

DISCLAIMER

ISACA has designed and created the *IS Audit/Assurance Program for Application Containers* (the "Work") primarily as an educational resource for IT audit professionals. ISACA makes no claim that use of any of the Work will assure a successful outcome. The Work should not be considered inclusive of all proper information, procedures and tests or exclusive of other information, procedures and tests that are reasonably directed to obtaining the same results. In determining the propriety of any specific information, procedure or test, IT audit professionals should apply their own professional judgments to the specific circumstances presented by the systems or information technology environment.

Reservation of Rights

ISACA®
1700 E. Golf Road, Suite 400
Schaumburg, IL 60173, USA

Phone: +1.847.660.5505

Fax: +1.847.253.1755

Support: support.isaca.org

Website: www.isaca.org

Provide Feedback:

www.isaca.org/application-containers

Participate in the ISACA Knowledge Center:

www.isaca.org/knowledge-center

Follow ISACA on Twitter:

www.twitter.com/ISACANews

Join ISACA on LinkedIn:

www.linkd.in/ISACAOfficial

Like ISACA on Facebook:

www.facebook.com/ISACAHQ

IS Audit/Assurance Program: Application Containers
ISBN 978-1-60420-629-6

Application Container Audit/Assurance Program

Column Name	Description	Instructions
Process Sub-area	An activity within an overall process influenced by the enterprise's policies and procedures that takes inputs from a number of sources, manipulates the inputs and produces outputs	To make the audit program manageable, it is recommended to break out the scope of the audit into sub-areas. The auditor can modify this field to entity-specific names and terms. ISACA has used the most commonly used terms as the basis to develop this audit program.
Ref. Risk	Specifies the risk this control is intended to address	This field can be used to input a reference/link to risk described in the entity's risk register or enterprise risk management (ERM) system or to input a description of the risk a particular control is intended to address.
Control Objectives	A statement of the desired result or purpose that must be in place to address the inherent risk in the review areas within scope	This field should describe the behaviors, technologies, documents or processes expected to be in place to address the inherent risk that is part of the audit scope. An IS audit manager can review this information to determine whether the review will meet the audit objectives based on the risk and control objectives included in the audit program.
Controls	The means of managing risk, including policies, procedures, guidelines, practices or organizational structures, which can be of an administrative, technical, management or legal nature	This field should describe in detail the control activities expected to be in place to meet the control objective. Control activities can be in roles and responsibilities, documentation, forms, reports, system configuration, segregation of duties, approval matrices, etc. An IS audit manager performing a quality control review must decide whether an auditor has planned to identify enough controls on which to base an assessment and whether the planned evidence is sufficiently objective.
Control Type	Controls can be automated (technical), manual (administrative) or physical. Automated/technical controls are things managed or performed by computer systems. Manual/administrative controls are usually things that employees can or cannot do. Physical controls include locks, fences, mantraps and even geographic specific controls.	Specify whether the control under review is automated, manual, physical or a combination. This information is useful in determining the testing steps necessary to obtain assessment evidence.
Control Classification	Another way to classify controls is by the way they address a risk exposure. Preventive controls should stop an event from happening. Detective controls should identify an event when it is happening and generate an alert that prompts a corrective control to act. Corrective controls should limit the impact of an event and help resume normal operations within a reasonable time frame. Compensating controls are alternate controls designed to accomplish the intent of the original controls as closely as possible when the originally designed controls cannot be used due to limitations of the environment.	Specify whether the control under review is preventive, detective, corrective or compensating. This information will be helpful when defining testing steps and requesting evidence.
Control Frequency	Control activities can occur **in real-time, daily, weekly, monthly, annually**, etc.	Specify whether the control under review occurs in real-time, daily, weekly, monthly, annually, etc. This information will be helpful when defining testing steps and requesting evidence.
Testing Step	Identifies the steps being tested to evaluate the effectiveness of the control under review	This field should describe in detail the steps necessary to test control activities and collect supporting documentation. The auditor can modify this field to meet entity-specific needs. ISACA has used a set of generic steps to develop this audit program. An IS audit manager may determine if the proposed steps are adequate to review a particular control.
Ref. Framework/Standards	Specifies frameworks and/or standards that relate to the control under review (e.g., NIST, HIPAA, SOX, ISO)	Input references to other frameworks used by the entity as part of their compliance program.
Ref. Workpaper	The evidence column usually contains a reference to other documents that contain the evidence supporting the pass/fail mark for the audit step.	Specify the location of supporting documentation detailing the audit steps and evidence obtained. An IS audit manager performing a quality control review must decide whether an auditor has tested enough controls on which to base an assessment and whether the obtained evidence is sufficiently objective to support a pass or fail conclusion.
Pass/Fail	Document preliminary conclusions regarding the effectiveness of controls.	Specify whether the overall control is effective (Pass) or not effective (Fail) based on the results of the testing.
Comments	Free format field	Document any notes related to the review of this Process Sub-area or specific control activities.

Application Container Audit/Assurance Program
Planning

Process Sub-area	Ref. Risk	Control Objectives	Controls	Control Type	Control Classification	Control Frequency	Testing Step	Ref. Framework/ Standards	Ref. Workpaper	Pass/Fail	Comments
Risk Analysis and Management		Risk associated with containerization is commensurate with enterprise risk tolerance.	A security review of containerization is performed on a periodic basis.				1. Interview security team members and/or review artifacts of risk identified during prior security reviews. 2. Ensure that security reviews are performed on the schedule identified by the enterprise.				
		The enterprise's IT risk and governance program addresses containers.	A container vulnerability assessment/remediation program has been designed and implemented.				1. Obtain an understanding of the enterprise's container vulnerability assessment/remediation program. 2. Confirm that all container application program elements are being performed and that any deficiencies noted have been remediated. Program elements may include, but are not limited to: • Formal deployment strategies with guidance on issue severity and corresponding remediation timeframes • Security scans that identify issues as well as monitor system performance (as an indicator of variance from baseline)				
Security Awareness and Training		Roles and responsibilities have been clearly defined so that organizational goals well as IT objectives are met. Note: Compared to the traditional application model, developers play a larger role in security in the application container environment. For example, because developers create images upon which containers are built, developers may perform patching, a function performed by IT Operations under a traditional application model.	Information security roles of IT Operations and of Developers have been defined and communicated.				1. Obtain the enterprise's policies and procedures related to application containerization. These may include, but are not limited to, the information security policy, application development policy, software development security policy and secure coding policy. 2. Review the policies/procedures to confirm that the roles of both IT operations and development have been defined. 3. Interview the appropriate personnel to determine how the enterprise communicates policy expectations. 4. Confirm that for applicable policies, the enterprise's communication protocol was followed. 5. If the enterprise has a policy acknowledgment process, confirm that IT operations and development personnel have active policy acknowledgments on file.				
		Data integrity is preserved through all phases of application containerization (planning, development, deployment, maintenance and destruction).	Segregation of duties has been created and is maintained in the application container environment.				1. Obtain an understanding of the enterprise's strategy to ensure segregation of duties. 2. Assess the adequacy of the strategy. Depending on the enterprise's application container environment (manual or automated), include the following points in the assessment. **Manual Environment** 1. Interview appropriate personnel to obtain an understanding of IT operations and development responsibilities. 2. Identify any potential lack of segregation in which one individual's access permissions are not limited based on the ability to perform multiple phases of a process; evaluate the potential risk. **Automated Environment** 1. Determine who has repository access to store images and who can download images. In Docker ®, for example, this access is reflected in the Docker trusted registry (DTR). 2. If defaults have been changed so that users can see containers other than their own, identify what user teams or groups have been created and the levels of control they have been assigned (e.g., read only or full control). Determine what resources the users have control over. In CoreOS rkt® (or Rocket®) for example, reliance is on the Linux control group feature for organizing processes into hierarchical groups and applying limits to the groups.				
		Roles and responsibilities have been clearly defined so that organizational goals well as IT objectives are met.	Developers are educated about their roles related to application security and are provided with training on security best practices.				Interview appropriate training personnel (or the chief information officer or chief information security officer) to identify application security training opportunities within the most recent twelve months. **Internal Training** • Review security awareness training materials and ensure that the training includes containers. • If the training is mandatory, confirm that IT operations and development staff have completed the training. • If the training is not mandatory, determine how IT operations and development staff obtain and maintain awareness of the enterprise's security best practices and expectations. **External Training** Confirm that any relevant external training is also reported by personnel to the enterprise (under professional development).				

Application Container Audit/Assurance Program
Development & Deployment

Process Sub-area	Ref. Risk	Control Objectives	Controls	Control Type	Control Classification	Control Frequency	Testing Step	Ref. Framework/ Standards	Ref. Workpaper	Pass/Fail	Comments
Host Operating System		The principle of least privilege is applied to ensure that users do not have access beyond what is required to perform their responsibilities.	Formal access management protocols (including but not limited to least privilege) exist and are enforced.				1. Obtain the enterprise's access management policy and review the portions related to application containers. 2. Obtain a list of people who were hired, transitioned from DevOps, or separated from the enterprise within the most recent twelve months. 3. From the population obtained above, make a selection of new hires, position transfers, and recent terminations. Confirm that access was granted/modified/revoked for new hires, position transfers and recent terminations respectively in a timely manner. For the sample of new hires and transfers, determine whether the access in place aligns with the access approved. 4. In assessing the appropriateness of access, confirm that users cannot log on directly to the host without authentication to an orchestration layer. **Note**: Namespace can also provide information about users and groups and the resources to which they have access.				
		The attack surface of the host OS is minimized by deactivation of unnecessary services. Risk of using a shared kernel, which is inherent in the application container infrastructure, is mitigated by reduction in the attack surface.	Unnecessary services are deactivated or a lightweight distribution is used to minimize the OS attack surface.				1. Through discussion with administrators, obtain an understanding of the enterprise's strategy for reducing the host OS's attack surface. 2. Confirm that the host OS runs only those services identified in the strategy. **Note:** Options for managing the host OS may include use of lightweight Linux® distribution or designation of a specific build for container hosting. Docker® and Rancher® OS are examples of specific builds. Specific OSs can contribute to a reduced attack surface by not having components such as libraries.				
Networking		The network type selected by the enterprise supports the enterprise's strategic objectives while ensuring confidentiality of network traffic.	Network traffic between containers on the same host is restricted.				1. Determine the type of network. In Docker®, for example, run the command to view existing container networks on the current Docker host. The command is `docker network ls`. Docker will return the network ID and the network type, host, bridge, overlay, or underlay. Network types include: • **None**—This type features a network stack, with no external network interface, but includes a loopback interface. • **Bridge**—This type has a namespace for each container provisioned inside a bridge. The bridge is provisioned on each host. Iptables with network address translation (NAT) map between each private container and the host's public interface. • **Overlay**—This type features a distributed network that covers (i.e., "overlays") host-specific networks. • **Underlay**—This type features hosts that interface directly with containers. 2. After determining the type of network, consider the enterprise's operations to assess any risk associated with the network type. For example, if container services need to be exposed to outside users, a bridge network is likely not the best choice.				
Container Runtime		Compromise of container runtime is mitigated to prevent exploitation of runtime software, which may allow a malicious actor to monitor container to container communications or even access the containers themselves.	The enterprise mitigates container runtime risk at launch and monitors risk on an ongoing basis.				1. Confirm that the default setting (which would allow individual containers to access each other and the host over the network) has been changed. 2. Identify the process that the enterprise uses to validate input. This process may reflect a manual procedure or an automated tool to ensure that inputs align with the enterprise's standard for acceptable input. Note that in addition to other inputs, the validation inputs may include configuration files, credentials and scripts during the build phase. 3. Determine whether the enterprise has adopted best practices related to container runtime as outlined by the Open Container Initiative® (OCI®).				
Images		Introduction of vulnerabilities (e.g., untrusted software or malware) via image downloading is minimized.	The enterprise designs and implements a process around trusted images.				1. Obtain information regarding the enterprise's strategy and protocols for trusted images. The enterprise's approach may include, but not be limited to, a central repository of trusted images, as well as control points within the development and deployment phases that ensure only trusted images are used. 2. Confirm that `DOCKER_CONTENT_TRUST` environment variable has been enabled so that only signed images can be used.				
			Sources of third-party images are verified prior to download and before use by the enterprise.				For a sample of recently downloaded images, confirm that verification took place prior to download. • For enterprises using Docker®, confirm that Docker® Content Trust is enabled. This feature supports whitelisting of repositories from authorized/trusted third parties. • For enterprises using rkt® (CoreOS Rocket®) or application container platforms, confirm that image scanning procedures are in place. **Note**: In addition to third-party container security solutions, there are open source solutions as well. • Determine if the enterprise has adopted best practices related to container images as outlined by the Open Container Initiative® (OCI®). **Note:** Malware detection should occur for images in registries as well as on the host.				
		Security risk is minimized through the design and implementation of image configuration protocols.	As part of risk analysis, the enterprise identifies image configuration protocols that align with the enterprise's risk tolerance.				1. Obtain documentation of the enterprise's image configuration protocol. 2. Identify tools used by the enterprise to enforce its protocol. 3. Interview appropriate personnel regarding compliance with the enterprise's image configuration protocol. Through these interviews, as well as review of documentation, confirm that alignment of configuration practices with protocol is assessed and ensure that any variances identified are resolved. 4. Confirm that digital signatures are verified prior to a container being placed in production.				
Registry		Confidentiality and security of images are maintained.	The enterprise ensures that its registries are conducted over a secure channel.				Observe settings and interview administrators to confirm that development tools, container runtime and orchestrators support encrypted channels for registries.				
			The enterprise uses tags on images to support identification and removal of outdated images.				1. Obtain information regarding how the enterprise identifies images that may need to be pruned because they have never been used or for other reasons. For example, enterprises that use Docker® may use the command of `docker image prune` to remove unused images. 2. Use the `docker image inspect` command to get detailed information on images. 3. Review the image information to determine if any images selected for testing are outdated according to the enterprise's criteria.				

Application Container Audit/Assurance Program Development & Deployment											
Process Sub-area	***Ref. Risk***	***Control Objectives***	***Controls***	***Control Type***	***Control Classification***	***Control Frequency***	***Testing Step***	***Ref. Framework/ Standards***	***Ref. Workpaper***	***Pass/Fail***	***Comments***
Orchestrator		Data confidentiality and security are maintained at the orchestrator level.	The principle of least privilege is applied to ensure that access is tailored to certain actions on certain containers. Note that in its support of multiple applications, a single orchestrator can involve different teams with different levels of sensitivity.				1. From the list of users obtained in host OS testing above, select a sample or users/groups and identify the access that users/groups have to certain containers, the host and images. 2. Given the roles of the users/groups, assess whether access is appropriate given the users'/groups' roles. 3. Identify whether any users/groups have access to production.				
Application Security During Development		Applications are less susceptible to exploitation.	The enterprise employs a robust application development process and/or application security development approach that adequately secures development (e.g., threat modeling).				1. Interview software development staff and/or review process documentation for any in-house developed software to ensure that software is developed using a robust process with a focus on building security into the process. 2. Confirm that patching is done in accordance with an enterprise wide accepted patching protocol or program. This confirmation should include review of patching for virtualization libraries (e.g., libcontainer, libvirt). Note that unlike traditional application models where patching is performed by IT operations, application container patching is part of the application build.				

Application Container Audit/Assurance Program
Maintenance

Process Sub-area	Ref. Risk	Control Objectives	Controls	Control Type	Control Classification	Control Frequency	Testing Step	Ref. Framework/ Standards	Ref. Workpaper	Pass/Fail	Comments
Secure Connections		Sensitive data related to secure connections between application container components (e.g., user name and password) are not compromised while stored in the image.	The enterprise uses its own sensitive data management system or an orchestration platform (e.g., Docker®, Swarm® or Kubernetes®) for sensitive data management (i.e., secrets management).				Confirm that sensitive data related to secure connections are stored outside of images.				
Hardening		A formal hardening program minimizes security risk.	The hardening program has a scanning component that supports identification of vulnerabilities in critical devices and applications.				1. Review the results of the most recent vulnerability assessment and/or penetration testing activities. 2. Ensure that testing is conducted periodically and that identified issues are addressed in a timely manner.				
			The hardening program incorporates application security scanning software (i.e., dynamic or static testing tools) or employs application-focused security testing to ensure that applications are secure.				1. Obtain a sample of recent application scanning results and/or testing results. 2. Review the results and ensure that artifacts are timely and reflect ongoing/periodic use; confirm that any identified issues are addressed.				
		Changes in container runtime are minimized so that changes take place during build rather than in production.	Immutable containers are used to prevent shell access to container images, which mitigates possible creation of a vulnerability.				Review the environment to ensure that containers as well as container elements are immutable.				
		Built-in security is leveraged in order to reduce the node attack surface.	Linux® features, such as Security Enhanced Linux (SELinux) and Secure Computing Mode (Seccomp), are used.				1. Determine the status of SELinux via the UNIX®/LINUX `getenforce` command. The `getenforce` command results indicate whether SELinux is enabled and whether its mode is enforcing or permissive. Please note that if hardened, the mode will be enforcing. 2. Determine if Seccomp is enabled by using the command: `$ grep CONFIG_SECCOMP=/boot/config-$(uname-r)CONFIG_SECCOMP=y`				
Container Destruction		Integrity is maintained throughout the life cycle of containers.	The enterprise mitigates risk associated with containers remaining active beyond their useful life (for example, a better program has been created).				1. Obtain an understanding of the enterprise's policy/protocol related to the end of life of application containers. The policy/protocol should include, but not be limited to, requests and approvals to destroy a container. 2. Because all data created over the container's lifetime are deleted when the container is destroyed, confirm how the enterprise ensures no data retention requirements are adversely affected (e.g., requirements related to compliance or electronic discovery). 3. If the enterprise uses Docker, determine which method of destruction is used: • `docker rm`—final destruction • `docker export`—archives the container's file system • `docker import`—retains the option of creating another container using the destroyed container				