AIA Group

Container Security Standard (ITSR.049)

Incorporating

All legal entities

Document Details

Document Name	Container Security Standard
Document Version	V1.1
Originating Business Function	Group Information Security
Policy Owner	Fields, Christopher – Digital Security
Primary Policy Contact Person	Praveen Karunakaran – Digital Security
Secondary Policy Contact Person	NA
Date of First Issuance	8-April-2019
Date of Last Approval	3-April-2020
Version Effective Date	3-April-2020
Notified to	CTO
Approved by	RCEs, Group CRO, Group General Counsel
Review Frequency	Once every 2 years or more frequently if needed
Next Review Date	3-April-2022
Document Type *Per Standard for Corporate Policy Governance*	Standard
Information Classification *Per Group Data Protection Standard*	Restricted
Related Policies and Standards	ITPR.001 Information Security Policy NIST.SP.800-190 ITPO.001 IT Change Management Policy ITSR.016 Network Security Standard ITSR.004 Cryptography Standard ITSR.020 Third Party Services Support Security Standard ITSR.034 Security Logging and Monitoring Standard ITSR.035 System Configuration Security Standard ITSR.037 User Identity and Access Management Standard ITRR.012 Technology Issue Management Process

VERSION CONTROL

Version	Amendments	Approval Date	Approved by
1.0	First Release	8-April-2019	RCEs, Group CRO, Group General Counsel
1.1.	Reviewed no change. Approved as per CPG review cycle.	3-apr-2020	RCEs, Group CRO, Group General Counsel

DISTRIBUTION LIST

TITLES
All employees

Contents

1. Objectives

The purpose of this Standard is to create a container security standard document that would assist in establishing a generic container security framework as well as provide security recommendations tailored towards the existing container specific toolset in AIA.

All reasonable effort must be taken to comply with as part of the Information Technology Security Policy. Any exceptions to this Standard must be approved through the Risk Acceptance Standard.

It is important to note that in addition to the mandatory requirements stated in this document, all supplementary actions required to comply with local government regulations under applicable jurisdictions must also be followed.

2. Scope and Definitions

2.1. Scope

This Standard applies to all AIA employees and business partners, third parties, and customers (employees and, as applicable, business partners, third parties and customers are hereafter in this document referred to as "Users") having access to AIA's information resources.

Control requirements defined in this Standard support and facilitate compliance with relevant laws and regulations of the countries in which AIA conducts business. Where local laws and regulations require controls that are more restrictive than those identified in this Standard, those more restrictive control requirements must also be complied with. In the event of a conflict, the control requirements of this Standard overrule the local laws and regulations, unless the local laws and regulations are more restrictive.

2.2. Definitions

Please refer to Appendix A of this Policy.

2.3. Assumptions & limitations

2.3.1. This Standard enables compliance with related legal and regulatory requirements of those countries that AIA have local presence. In the event of contradiction or conflict between the two, the more restrictive one shall be fully complied.

2.3.2. Local Information Security or IT Security is responsible for monitoring compliance of all mandatory security requirements and assisting to implement appropriate measures.

2.3.3. This Standard shall be reviewed once every 2 years or frequently if needed.

3. Roles and Responsibilities

Functional Roles	Responsibilities
Application Department Head	• Ensure team members are conformed to this standard when using a container
Local Information Security Team	• Coordinate container security scanning for respective Business units
Group Information Security Team	• Build secure container repository • Monitor container scanning • Review and assess the effectives of container security

4. Container Security

4.1. Securing the Container Images

The following minimum controls must be implemented for all the core components of container technologies—images, registries, orchestrators, containers, and host Oss

4.1.1. Container Image Vulnerabilities

To detect vulnerabilities within containers, container technology-specific vulnerability management tools and processes must be used.

Tools that take the pipeline-based build approach and immutable nature of containers and images into their design to provide more actionable and reliable results must be used. Key aspects of effective tools and processes include:

a. Integration with the entire lifecycle of images, from the beginning of the build process, to the container registries and runtime.

b. Visibility into vulnerabilities at all layers of the image, not just the base layer of the image but also application frameworks and custom software.

c. Policy-driven enforcement; only images that meet the organization's vulnerability and configuration policies must be allowed to progress by creating "quality gates" at each stage of build and deployment process.

4.1.2. Container Image Configuration.

Standard tools and processes must be used to validate and enforce compliance with secure configuration best practices.

Tools and processes that must be adopted include:

a) Validation of image configuration settings, including vendor recommendations and third-party best practices

b) Ongoing, continuously updated, centralized reporting and monitoring of image compliance state to identify weaknesses and risks

c) Enforcement of compliance requirements by optionally preventing the running of non-compliant images.

d) Use of base layers from trusted sources only, frequent updates of base layers, and selection of base layers from minimalistic technologies

e) SSH and other remote administration tools designed to provide remote shells to hosts should never be enabled within containers

f) Containers must be run in an immutable manner

g) All remote management of containers must be done through the container runtime APIs, which can be accessed via orchestration tools, or by creating remote shell sessions to the host on which the container is running.

h) All the images must be monitored for embedded malware. And the monitoring solution must include the use of malware signature sets and behavioural detection heuristics based largely on actual “in the wild” attacks

4.1.3. Securing storage of container secrets

a. Secrets must be stored outside of images and provided dynamically at runtime as needed

b. Use a standard orchestration tool to manage secure storage of secrets.

c. Container secrets must always be encrypted at rest and in transit as per Data Protection standard.

4.1.4. Protection from untrusted container images

a. Container images used must be fetched from verified authentic sources.

b. A set of trusted images and registries must be maintained to ensure that only images from this set must be allowed to run in their environment.

4.2. Securing Container Registry

4.2.1. Securing connection to registries

a. Development tools, orchestrators, and container runtimes must be configured to only connect to registries over encrypted channels.

b. All data pushed to and pulled from a registry must occur between trusted endpoints and is encrypted in transit.

4.2.2. Verify Container images in registries

a. Container Images in registries must be checked for any unsafe and vulnerable images by using a container aware scanning tool. Any unsafe or vulnerable image must be removed from registry.

b. Stale images must be removed from the registries.

4.2.3. Authentication and Authorization

a. All access to registries that contain proprietary and / or sensitive images must require authentication.

b. Any write access to a registry must require authentication to ensure that only images from trusted entities can be added to it.

Access control must be on need to know basis. Access must be restricted to push images to the specific repositories authorized for, rather than being able to update any repository.

4.3. Container Runtime Monitoring and Protection

4.3.1. Protecting Runtime Software

The container runtime must be carefully monitored for vulnerabilities, and when problems are detected, they must be remediated quickly.

4.3.2. Unbounded network access from containers

Containers must not be allowed to communicate with one another unless authorized to.

4.3.3. Securing Container Runtime configuration

Automate security and compliance with container runtime configuration

4.3.4. Container App protection

Container app must be verified to prevent and detect anomalies at runtime including events such as:

a. Invalid or unexpected process execution,

b. Invalid or unexpected system calls,

c. Changes to protected configuration files and binaries,

d. Writes to unexpected locations and file types,

e. Creation of unexpected network listeners,

f. Traffic sent to unexpected network destinations, and

g. Malware storage or execution.

4.3.5. Container Environment Protection

a. There must be separate environments for development, test, production, and other scenarios, each with specific controls to provide role-based access control for container deployment and management activities.

b. Container Images must only have components required for applications to function as expected.

c. All container creation must be associated with individual user identities and logged to provide a clear audit trail of activity.

d. Enforce baseline security requirements for vulnerability management and compliance prior to allowing an image to be run.

4.4. Securing Host OS

4.4.1. Minimize the Attack Surface

a. Hosts must be continuously scanned for vulnerabilities and updates applied quickly, not just to the container runtime but also to lower-level components such as the kernel

b. Container-specific OS must be used to host containers and ensure that unnecessary services are disabled.

4.4.2. Securing Kernel

a. Keep containerized workloads isolated to container-specific hosts

b. Do not mix containerized and non-containerized workloads on the same host instance.

4.4.3. Securing Host OS Components

a. Validate the versioning of components provided for base OS management and functionality.

b. Host OS must be operated in an immutable manner with no data or state stored uniquely and persistently on the host and no application-level dependencies provided by the host.

4.4.4. User Access Rights

All authentication to the OS must be audited, login anomalies must be monitored, and any escalation to perform privileged operations must be logged

4.4.5. Host File System Tampering protection

a. Containers must run with the minimal set of file system permissions required.

b. Containers must not be able to mount sensitive directories on a host's file system, especially those containing configuration settings for the operating system.

4.5. Orchestrator Security

4.5.1. Access control

Access control must be in line with AIA Identity and Access Management standard.

4.5.2. Maintain Orchestrator node trust

a. Maintain secure orchestrator configuration for trust between the nodes

b. Must have controls in place to prevent unauthorized hosts joining the cluster and running containers

c. Communications between the orchestrator and DevOps personnel, administrators, and hosts must be encrypted and authenticated.

5. Exemptions

All requirements stipulated in this standard are mandatory. Any deviations from this Standard must be approved by Information Security through the Technology Issue Management Process.

5.1. Breach Management and Escalation

Any breaches of any requirements in this document prior to authorisation must be escalated to Local BU Information Security for Local BU, Group Information Security for Group functions and document owner.

Group and BU required to go through the Technology Issue Management Issue Management process with Information Security.

5.2. Monitoring, Review and Amendments

Local Information Security and Controls function is responsible for monitoring compliance of all mandatory security requirements and assisting to implement appropriate measures. Instances of non-compliance required to be escalated and get approval by Information Security through TIM process. This document will be reviewed on at least once every 2 years or frequently if needed. Minor amendments or cosmetic changes such as typo, text or table alignment and formatting that do not have significant impact to the requirements discussed in this document; do not require re-approvals from appointed parties mentioned in Document Details.

5.3. Delegation of Authority and Other Administrative Matters

Controls functions responsible to ensure the execution of this document requirement. No delegation of authority to any parties. Latest effective version of this document must reside on Corporate Policy Portal (“CPP”) as per CPG standard. Any inquiries of this document must be made to the document owner or primary contact as listed.

6. Approvals

This document is approved by the RCEs, Group CRO and Group General Counsel. Notification to CTOO.

7. Appendix A - Glossary

“must”	The use of the word “must” indicates a mandatory requirement.
“should”	The use of the work “should” indicates a requirement for good practice, which to be implemented whenever possible.
“may”	The use of the word “may” indicates a desirable requirement.
AIA	AIA Group Limited and each of its subsidiaries and branches
Business Unit	Business entity that is wholly-owned by AIA Group Limited or a subsidiary of AIA. Examples are: AIA Australia, AIA Singapore/Brunei, AIA China, AIA Hong Kong/Macau, AIA Financial, AIA Korea, AIA Insurance Lanka PLC, AIA Malaysia, AIA New Zealand, AIA PT, AIA Thailand, AIA Taiwan, AIA Vietnam, AIA Shared Services Hong Kong, TSS (BJ), TSS(GZ), OSS(ML), Philam Life

AIA Group

Container Security Standard

Version 1.1