

Technical and Organizational Measures (TOMs)

Aklivity, Inc.

Version: 1.0

Effective Date: January 6, 2025

Contact: security@aklivity.io

1. Introduction

This document outlines the technical and organizational measures implemented by Aklivity, Inc. to ensure the confidentiality, integrity, and availability of its software and services. Aklivity provides a self-managed, container-based software solution (e.g., Zilla, Zilla Plus) that customers deploy within their own environments. As such, Aklivity does not operate or access customer data or infrastructure by default.

2. Scope

These measures apply to:

- The software artifacts delivered by Aklivity (e.g., Docker images).
- The development, testing, and release pipelines used to build and distribute the software.
- Aklivity's internal systems and processes related to secure software development and support.

3. Data Handling Responsibility

Customer-Controlled Environment:

Aklivity's software runs entirely within the customer's own infrastructure. Aklivity does not:

- Access customer data.
- Transmit or process customer data in a managed service.
- Have persistent connections to customer systems.

Customers are responsible for:

- Securing their deployment environment.
- Managing access, monitoring, and data governance.

4. Technical Measures

4.1. Secure Software Development Lifecycle (SSDLC)

- Use of Git-based version control with access control and code review policies.
- Static Application Security Testing (SAST) and dependency scanning integrated into CI/CD pipelines.
- Mandatory peer reviews for all major code changes.
- Security requirements integrated into design and architecture phases.

4.2. Container Image Security

- All containers are built from minimal, hardened base images.
- Regular vulnerability scans (e.g., Trivy, Snyk) during CI.

4.3. Supply Chain Integrity

- Software dependencies are pinned and verified.
- External dependencies are sourced from trusted repositories.
- SBOM (Software Bill of Materials) generated and published with each release.

4.4. Artifact Delivery

- Docker images are published to a secure, authenticated container registry.

5. Organizational Measures

5.1. Access Management

- Role-based access controls (RBAC) on all internal systems.
- Multi-factor authentication (MFA) required for access to code repositories, build systems, and cloud infrastructure.
- Least privilege access principle enforced.

5.2. Security Awareness

- Annual security training for all employees.
- Secure coding workshops and role-specific security training for engineering teams.

5.3. Incident Response

- Formalized incident response plan for internal systems and software supply chain threats.
- No customer data involved unless explicitly escalated during support interactions.

6. Support & Escalations

When support access is required (e.g., for a customer-reported issue):

- Customers may voluntarily share logs or configurations.
- Any shared data is handled under strict confidentiality.

7. Customer Responsibilities

Customers deploying Aklivity software are responsible for:

- Applying security updates promptly.
- Isolating the containerized deployment as per best practices (e.g., using firewalls, role-based service accounts).
- Enabling audit logging and monitoring appropriate for their environment.

8. Compliance and Auditing

While Aklivity does not process customer data, we maintain internal compliance with:

- SOC 2-aligned internal controls (in-progress).
- Secure-by-design engineering principles.
- Transparency in supply chain and build pipeline integrity.

9. Contact

For security-related questions, disclosures, or compliance requests, contact:

 security@aklivity.io