Security Strategy for a Cloud Infrastructure

Security Strategy

Author: Tshepo Mphelane

Table of Contents

[1 Introduction 2](#_Toc198377316)

[2 Encryption Methods 2](#_Toc198377317)

[2.1 Encryption of data at rest 2](#_Toc198377318)

[2.2 Data-in-Transit Encryption 2](#_Toc198377319)

[2.3 Key Management 2](#_Toc198377320)

[3 Access Control Policies 3](#_Toc198377321)

[3.1 Identity and Access Management (IAM) 3](#_Toc198377322)

[3.2 Authentication 3](#_Toc198377323)

[3.3 Credential Management 3](#_Toc198377324)

[4 Compliance Measures 3](#_Toc198377325)

[4.1 Regulatory Compliance 3](#_Toc198377326)

[4.2 Security Frameworks 3](#_Toc198377327)

[4.3 Auditing and Monitoring 4](#_Toc198377328)

[5 Incident Response and Breach Handling Plan 4](#_Toc198377329)

[5.1 Preparation 4](#_Toc198377330)

[5.2 Detection and Analysis 4](#_Toc198377331)

[5.3 Containment, Eradication, and Recovery 4](#_Toc198377332)

[5.4 Post-Incident Activities 4](#_Toc198377333)

[6 Continuous Improvement 5](#_Toc198377334)

[7 References: 5](#_Toc198377335)

# Introduction

This document outlines a comprehensive security strategy for securing a cloud-based infrastructure that is hosted in Azure Kubernetes Service (AKS). It is designed to protect data, applications, and systems from unauthorized access and other security threats while ensuring compliance with relevant regulations and standards.

# Encryption Methods

## Encryption of data at rest

* **Encryption Algorithms**: AES-256 encryption for all stored data.
* **Storage Services**:
  + Enable encryption on cloud storage service like Azure Blob Storage.
  + Use cloud provider-managed keys (default) or customer-managed keys (CMKs) with Key Management Service (KMS) for higher control.
* **Database Encryption**:
  + Enable Transparent Data Encryption (TDE) for relational databases.
  + Encrypt NoSQL and big data platforms using built-in encryption or third-party solutions.

## Data-in-Transit Encryption

* Enforce TLS 1.2 or higher for all data transmission.
* Use HTTPS with certificates managed through a secure Certificate Authority (CA).
* Apply mutual TLS (mTLS) for inter-service communication in microservices architectures.

## Key Management

* Utilize a centralized Key Management System like Azure Key Vault.
* Rotate encryption keys regularly and automatically where supported.
* Limit key access using IAM policies and audit all key usage.

# Access Control Policies

## Identity and Access Management (IAM)

* Grant users and services the minimum permissions necessary.
* **Role-Based Access Control (RBAC)**: Define roles based on job functions and assign users accordingly.
* **Just-in-Time Access**: Temporary elevation of privileges *only* when necessary, with approval workflows.

## Authentication

* Use multi-factor authentication (MFA) for all users, especially administrators.
* Integrate with identity providers (Azure AD, Okta) using SAML for Single Sign-On (SSO).

## Credential Management

* Prohibit the use of hardcoded credentials in code.
* Use secret management tools (e.g. HashiCorp Vault, Azure Key Vaults) for storing API keys and secrets.

# Compliance Measures

## Regulatory Compliance

* **GDPR**: Implement data protection principles, data subject rights, and breach notification mechanisms.
* **PCI-DSS**: Secure payment data through tokenization, network segmentation, and strict access controls.

## Security Frameworks

* Align with industry best practices such as:
  + NIST Cybersecurity Framework (CSF)
  + ISO/IEC 27001
  + CIS Critical Security Controls

## Auditing and Monitoring

* Enable centralized logging like Azure Monitor.
* Use SIEM tools (Splunk, Datadog) to correlate and analyze security events.
* Schedule periodic audits and vulnerability assessments.

# Incident Response and Breach Handling Plan

## Preparation

* Maintain an up-to-date **Incident Response Plan (IRP)**.
* Define incident roles and responsibilities (incident commander, communications lead).
* Establish secure communication channels for incident response.

## Detection and Analysis

* Set up automated alerting for:
  + Irregular login attempts
  + Privilege escalations
  + Data exfiltration attempts
* Regularly review security logs and threat intelligence feeds.

## Containment, Eradication, and Recovery

* **Containment**: Isolate affected systems to prevent spread.
* **Eradication**: Remove root cause (malware, misconfigurations).
* **Recovery**: Restore systems from trusted backups, validate data integrity, and monitor for recurrence.

## Post-Incident Activities

* Conduct a **post-mortem analysis** and document lessons learned.
* Update incident response procedures and controls based on findings.
* Communicate transparently with affected stakeholders and comply with breach notification laws.

# Continuous Improvement

* Implement **DevSecOps** practices to integrate security into CI/CD pipelines.
* Conduct regular **penetration testing** and red team exercises.
* Review and update the security strategy quarterly or after major system changes.

# References:

**Encryption Methods**

<https://learn.microsoft.com/en-us/azure/security/fundamentals/encryption-overview>

**Access Control:**

<https://learn.microsoft.com/en-us/entra/fundamentals/introduction-identity-access-management>

**Data Compliance:**

<https://gdpr.eu>

<https://www.iso.org/isoiec-27001-information-security.html>

**Incident Response:**

<https://learn.microsoft.com/en-us/security/operations/incident-response-overview>