Systematization of Knowledge on Hardening Kubernetes through Native Configurations

LABCIB

Master in Informatics Engineering - 2024/2025

Porto, November 17, 2024

1190830 - Luís Correia

1190974 - Pedro Lemos

1191526 - Pedro Moreira

Version 2, 2024-10-17

Revision History

Revision	Date	Author(s)	Description
1	2024-11-15	Pedro Moreira	Initial version and structure
2	2024-10-17	...	Extended description

Contents

List of Figures

1 Introduction

[Description, considering the project, of what under individual responsibility]

2 Objectives and Scope

[Description, for what under individual responsibility, that inform the goal, the questions related to functional correctness, metrics used and their values, with partial response to questions, and goal achievement analysis. Explicitly mention the tool report(s).]

3 Methodology

[Description, for what under individual responsibility, that inform the goal, the questions related to maintainability, metrics used and their values, with partial response to questions, and goal achievement analysis. Explicitly mention the tool report(s).]

4 Background

Explicar o que é o Kubernetes e como usa o etcd (isto é importante). Falar também brevemente sobre o que são Secrets (porque depois vamos referir como é que os protegemos melhor). Relativamente ao Kubernetes parece-me interessante falar do básico, ou seja o que são master e worker nodes, quais são os componentes (API server, kubelet, essas coisas que se encontram facilmente na documentação deles). Não precisa de ser muito extensivo mas convém dar algum contexto para depois sabermos o que estamos a dizer

5 Kubernetes Native Configurations for Hardening

Having introduced what Kubernetes is, we are now able to delve into the main purpose of this report, which is to systematize what built-in features and configurations of this technology we can use to improve a cluster's security posture and resistance to cyber-attacks. This chapter will be split into various sections, each concerning an area of relevance when it comes to Kubernetes Hardening. In each section, a context regarding the security improvements that are available will be presented, and consequently the measures through which one can implement these improvements will also be showcased.

5.1 Data Encryption

As we've seen in the context section of this report, Kubernetes stores Secrets and their data on the etcd backend database without any form of encryption [1] - despite encoding them with Base64, which does not constitute encryption as the plain-text contents are directly obtainable. If an attacker is able to access either the API server or the etcd database directly, he will be able to obtain the plain-text contents of the secrets that are managed by that cluster. One can easily picture the compromise that this can cause, as secrets usually contain business-critical credentials that are used by services running in Kubernetes. For instance, an application Pod might need to access a database, and thus read the connection string and database user credentials from a secret. If an attacker could obtain the content of this secret, he too would at least have the credentials to access the database.

Fortunately, Kubernetes provides a series of configurations that can be applied to encrypt Secrets - and other resources - at rest. This is done through the configuration of some resources on the cluster's API server. Notably, Kubernetes defines the `EncryptionConfiguration` Custom Defined Resource (CRD), which allows practitioners to customize what resources shall be encrypted and through what mechanism (also known as provider) [2].

The flow of actions is as follows: a user or application creates a Secret by sending an API call to the Kubernetes API; the API checks its `EncryptionConfiguration` file to evaluate what encryption provider to use and which key to use for that provider; if a provider is specified, as well as the key to use, the API server uses both to encrypt the provided data; then, this encrypted data is stored in etcd. For the decryption, the API retrieves the secret from etcd and uses the aforementioned `EncryptionConfiguration` to decrypt the contents.

5.2 API Server Configurations

5.3 Network Policies

5.4 RBAC and Authentication

6 Complementary Tooling

[Description, for what under individual responsibility, that inform the goal, the questions related to security, metrics used and their values, with partial response to questions, and goal achievement analysis. Explicitly mention the tool report(s).]

7 Analysis and Best Practices

[Description, for what under individual responsibility, that inform the goal, the questions related to architectural compliance, metrics used and their values, with partial response to questions, and goal achievement analysis. Explicitly mention the tool report(s).]

8 Conclusions

...

Bibliography

[1] The Linux Foundation. Secrets. en. Section: docs. 2024. URL: https://kubernetes.io/docs/concepts/configuration/secret/ (visited on 11/17/2024).

[2] The Linux Foundation. Encrypting Confidential Data at Rest. en. Section: docs. 2024. URL: https://kubernetes.io/docs/tasks/administer-cluster/encrypt-data/ (visited on 11/17/2024).