SECURE SECRETS MANAGEMENT IN KUBERNETES

Kadatskaja O., Driss Alaoui Soulimani

V.V. Popovsky Dep.Engineering Infocommunication HNURE,
Ukraine.

E-mail: olha.kadatska@nure.ua,
sulimani.driss.alaui@nure.ua

Abstract

With the rise of cloud-native applications, secure secrets management is a critical cybersecurity concern. Mismanagement of API keys, tokens, and credentials can lead to severe breaches. This article discusses the importance of managing secrets in Kubernetes clusters, highlighting the benefits of integrating external secrets management solutions like AWS Secrets Manager. We examine theoretical foundations and real-world scenarios that reveal the risks and challenges organizations face in managing secret.

Cloud-native applications heavily rely on sensitive information, such as credentials, tokens, and encryption keys, to function. Kubernetes, a widely adopted cloud platform, requires a secure approach to managing these secrets. Mismanagement of secrets can lead to data breaches, unauthorized access, and compliance issues. This article explores how secrets are handled in Kubernetes and emphasizes the need for secure, external management solutions like WS Secrets Manager.

Theoretical Background Concept of Secrets in Kubernetes In Kubernetes, secrets are objects that store sensitive data such as passwords, API tokens, and encryption keys. However, Kubernetes secrets are only base64-encoded by default, offering minimal protection. If a cluster is compromised, these secrets can be exposed, making encryption and robust management essential. Kubernetes offers two main approaches for managing secrets: internal and external.

Internal Secrets: These are stored locally within the Kubernetes cluster. While suitable for small environments, internal secrets pose risks due to weak encoding and a lack of automated rotation mechanisms.

External Secrets: Managed by external services like AWS Secrets Manager, external secrets provide stronger security through encryption, centralized management, and automatic rotation.

The following table compares key features of internal and external secret management (table 1).

Table 1. Key features of internal and external secret management

Feature	Internal Secrets	External Secrets (e.g., AWS Secrets Manager)
Storage Location	Stored within the Kubernetes cluster.	Managed remotely by AWS, separate from the cluster.
Security	Base64-encoded, vulnerable to misuse.	Encrypted, centralized, and auditable.
Rotation	Requires manual intervention.	Automated rotation supported by AWS.
Scalability	Limited to local clusters.	Easily scalable across multiple clusters.
Access Control	Kubernetes RBAC-based access.	Fine-grained control via AWS IAM.
Compliance & Auditing	Minimal auditing capabilities.	Full audit logs and compliance controls.

Key Challenges in Secret Management:

1. Hardcoded Secrets in Version Control: A common mistake is embedding secrets directly into source code. In a notable 2018 breach, API keys were exposed in a public GitHub repository, allowing attackers to access critical infrastructure.

Lesson: Secrets should never be stored in version control, even with .gitignore files.

2. Outdated Secrets: Without automated rotation, credentials can become outdated, creating security vulnerabilities. A financial institution faced significant compliance issues when exposed credentials remained unchanged for too long. *Lesson*: Automated secret rotation is crucial for maintaining security.
3. Insecure Kubernetes Secrets: A cloud service provider experienced a data breach when its internal Kubernetes secrets, which were only base64-encoded, were compromised. *Lesson*: Native Kubernetes secret management is inadequate for high-security environments and should be supplemented with external solutions.

Risks and challenges in secret management organizations face several challenges when managing secrets in cloud-native environments:

1. Centralized Management: Managing secrets across multiple clusters can become cumbersome without a centralized solution.
2. Access Control: Proper role-based access is critical to ensuring that only authorized users and services have access to secrets.
3. Secrets Rotation: Regular secret rotation is essential to maintaining security, but it can be error-prone and time-consuming without automation.
4. Auditability: Tracking who accesses secrets is critical for security and compliance, particularly in regulated industries such as finance or healthcare.

Real-World Case Studies:

1. Hardcoded Secrets: The 2018 breach, where a company's API keys were exposed on GitHub, exemplifies the risk of embedding secrets in source code. *Lesson*: Secrets should be stored securely outside the codebase, ideally in an external secrets manager.
2. Failure to Rotate Secrets: A financial institution discovered that its failure to regularly rotate secrets created security vulnerabilities, leading to compliance issues. *Lesson*: Automated secret rotation is necessary to prevent the use of outdated credentials.
3. Kubernetes Secrets Vulnerability: A cloud provider's breach demonstrated that Kubernetes' default base64 encoding for secrets is insufficient, particularly for high-security environments. *Lesson*: External solutions like AWS Secrets Manager offer greater security through encryption and centralized control.

Conclusion

Effective secret management is crucial for any organization using cloud-native platforms like Kubernetes. Internal secret management mechanisms, such as Kubernetes Secrets, are often insufficient due to their lack of encryption, auditing, and automated rotation. External solutions, such as AWS Secrets Manager, significantly enhance security, offering encryption, centralized management, and scalability. By adopting these practices, organizations can mitigate risks, improve compliance, and strengthen their overall security posture.

References

1. Grout, V., & Clayton, R. (2020). *The Challenges of Secure Secret Management in Cloud-Native Applications*. Journal of Cybersecurity, 5(2), 100-115. https://doi.org/10.1016/ j.jcs.2020.100123.
2. Kang, Y., & Kim, H. (2021). *Managing Secrets in Cloud-Native Architectures: A Comparative Study of Kubernetes and External Solutions*. Proceedings of the International Conference on Cloud Computing, 102-110. https://doi.org/10.1109/ICCC.2021.9341234.
3. Amazon Web Services (2021). *Best Practices for Managing Secrets in AWS Secrets* Manager. AWS Whitepapers. Retrieved from https://docs.aws.amazon.com/secretsmanager/ latest/userguide/best-practices.html