An Evaluation of Kubernetes Security Mechanisms for DoS Prevention: A Comparative Analysis of Calico Network policies, K-Rail & Open Policy Agent, and Consul Rate Limiting on AWS EKS

Mia Kuric
School of Enterprise Computing and Digital Transformation, TU Dublin, Ireland
X00218710@mytudublin.ie

Introduction

Kubernetes has become the dominant orchestration platform for cloud-native applications due to its scalability, automation, and portability. However, its widespread adoption and distributed architecture have also made it a prime target for Denial-of-Service (DoS) attacks. DoS attacks aim to exhaust system resources such as CPU, memory, or network bandwidth, leading to service degradation or complete unavailability. In Kubernetes environments, these attacks are particularly impactful because of shared cluster resources, complex service dependencies, and extensive east–west traffic between microservices. This project evaluates the effectiveness of three widely used Kubernetes security mechanisms operating at different layers of the stack: Calico Network Policies (network layer), K-Rail with Open Policy Agent (OPA) (admission and configuration layer), and Consul Rate Limiting (application layer). Controlled DoS experiments were conducted in both AWS Elastic Kubernetes Service (EKS) and Minikube environments to analyse their impact on system availability, performance, and resource consumption. The results provide practical insights into how defence-in-depth strategies can improve Kubernetes resilience against DoS attacks.

Objectives

- Evaluate the effectiveness of Kubernetes security mechanisms in mitigating Denial-of-Service (DoS) attacks
- Analyse the performance impact of Calico Network Policies, K-Rail with OPA, and Consul Rate Limiting when deployed individually
- Assess how deployment environment (AWS EKS vs. Minikube) influences DoS resilience
- Develop practical recommendations for implementing multi-layer Kubernetes security strategies that balance protection and performance

Research Question

How effective are Calico Network Policies, K-Rail & Open Policy Agent, and Consul's Rate Limiting in mitigating DoS attacks in cloud-based Kubernetes environments?

Performance Impact and Security Trade-Offs

Performance Impact
Calico network policies imposed very little additional CPU or memory usage and were effective at filtering unauthorised traffic at the network layer. However, once traffic was allowed, Calico offered no protection against application-layer attacks such as HTTP flooding.

K-Rail combined with OPA introduced almost no runtime overhead, since policy enforcement occurs during admission rather than at execution time. Although it does not actively stop live attack traffic, it helped limit the potential impact of attacks by enforcing safer configurations and resource constraints.

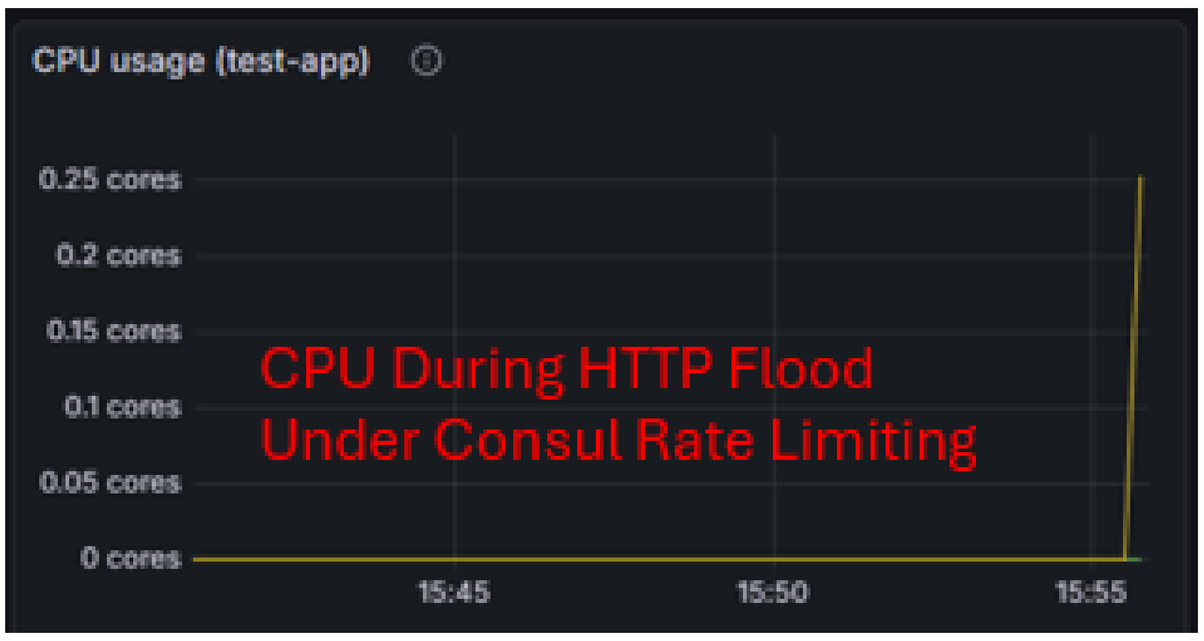
Consul rate limiting delivered the most effective defence against HTTP flood attacks. This came at the cost of increased CPU and memory consumption, mainly due to the use of Envoy sidecar proxies, which became more noticeable under sustained high request rates.

Security Trade-offs
Mechanisms with low resource overhead tend to offer protection over a narrower attack surface
Controls operating at the application layer provide stronger resistance but require greater system resources
No single solution was sufficient to fully protect against all forms of DoS attacks

Key insight: Security effectiveness and performance overhead are inherently linked; trade-offs are unavoidable.

Topic Overview

Mechanism	Effective Against	Ineffective Against
Syncookies (Linux)	Partial SYN flood mitigation	High CPU pressure
Calico	Block unauthorized L3/L4 traffic	Allowed HTTP floods
K-Rail & OPA	Prevent unsafe pod configs	Runtime DoS
Consul Rate Limiting	HTTP floods (L7)	SYN floods



Conclusions and Future Work

This study demonstrates that Denial-of-Service (DoS) resilience in Kubernetes cannot be achieved through only one security mechanism. Calico Network Policies provide efficient early packet filtering, K-Rail with OPA strengthens cluster security posture by preventing insecure deployments, and Consul Rate Limiting delivers robust application-layer protection. When combined, these mechanisms form a defence-in-depth strategy that significantly improves availability and stability under DoS conditions.

Future research should extend this work by:
- Evaluating additional service meshes such as Istio or Linkerd
- Testing larger Kubernetes clusters
- Investigating adaptive, telemetry-driven DoS mitigation using eBPF

QR Code for Recording