Kubernetes Threat Model

Denial of Service

Denial of Service (DoS) attacks aim to disrupt the normal functioning of a service by overwhelming it with a flood of illegitimate requests. In a Kubernetes environment, DoS attacks can affect both the cluster and the applications running within it, leading to resource exhaustion and service downtime.

Kubernetes provides various mechanisms to mitigate the risk of DoS attacks, including resource quotas, limits, and network policies. By implementing these controls, administrators can prevent a single user or application from consuming excessive resources and ensure the stability and availability of the cluster.

RealLife Example:

Imagine a powerful kingdom facing a siege where the enemy aims to overwhelm the defenses and prevent legitimate citizens from entering. In Kubernetes, DoS attacks target the cluster's resources, overwhelming them and rendering applications inaccessible to authorized users.

Key Concepts

1. **Resource Quotas and Limits**
   - Resource Quotas: Limit the total amount of resources (CPU, memory, storage) that a namespace can consume.
   - Resource Limits: Define the maximum amount of resources that a pod or container can use.
2. **Network Policies**
   - Control the flow of traffic between pods and external endpoints.
   - Can be used to limit the impact of network-based DoS attacks.
3. **Horizontal Pod Autoscaler (HPA)**
   - Automatically scales the number of pod replicas based on observed CPU utilization or other select metrics.
   - Helps to handle increased load and mitigate the impact of DoS attacks.

4. **Pod Disruption Budgets (PDB)**
    - Ensure a minimum number of pods remain available during voluntary disruptions, such as maintenance or scaling events.
5. **Rate Limiting**
    - Limit the rate of incoming requests to services to prevent overloading.

Security Best Practices

1. **Implement Resource Quotas and Limits**
    - Set resource quotas at the namespace level to control the overall resource usage.
    - Define resource limits for individual pods and containers to prevent resource hogging.
2. **Use Network Policies**
    - Define network policies to restrict traffic flow and protect sensitive services from external attacks.
    - Isolate critical services to minimize the impact of potential DoS attacks.
3. **Enable Horizontal Pod Autoscaling**
    - Configure HPA to automatically scale pods in response to increased load.
    - Set appropriate scaling policies to balance between performance and resource usage.
4. **Monitor and Alert**
    - Use monitoring tools (e.g., Prometheus, Grafana) to track resource usage and detect anomalies.
    - Set up alerts to notify administrators of potential DoS attacks.
5. **Rate Limiting**
    - Implement rate limiting at the ingress controller or application level to prevent excessive request rates.