TIGERA

Managing Application Connectivity in the Enterprise

Achieving Compliance and Organizational Alignment with Hierarchical Security Policy

Overview

Development teams are increasingly adopting microservices-based approaches to enhance their organization's speed and agility. As they do so, many are quickly realizing that responsibility for application management must reside with those who best know what's being managed: application owners. These responsibilities must include network security, but this raises a key concern for compliance and information security administrators: How can network security policy be decentralized among application owners, while still ensuring compliance with the organization's overarching security objectives, such as preserving information integrity and preventing breaches?

This white paper describes a policy strategy that enables administrators to meet both objectives.

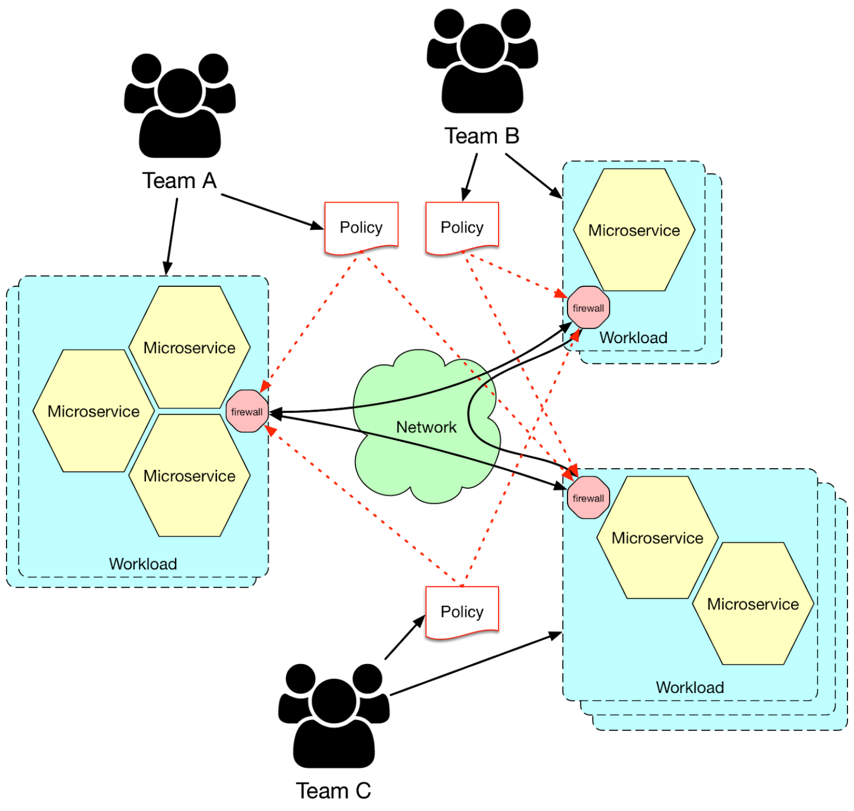
Table of Contents

Introduction: The Complex Side Effects of Simple Microservices

There's a microservices revolution taking place. Through microservices, organizations are delivering applications that can evolve rapidly, and reliably process larger volumes of data and requests. Microservices also features a distributed architecture that empowers teams to build smaller and simpler components that are focused on a specific domain and independently deployable. This enables teams to manage development lifecycles autonomously. This simplification does not come without side-effects.

Moving Component Policy to the Network

Component interactions used to be implemented as in-process library function calls. Today, they are run as out-of-process web service and other network-based calls. This difference has significant security implications. This means that policies governing information flow can no longer be enforced primarily in-process by libraries and runtime frameworks. Instead, policies must be applied at the service or network boundary.



Dealing with Constant Change

Microservices approaches have additional implications as well. By moving the component interaction interface out of the runtime environment and into the network, microservices fuel a dramatic increase in the diversity and volume of service-to-service traffic in the data center. Most often, applications and platforms built on microservices are packaged and deployed as a collection of collaborating containers. The scaling and dynamic scheduling of these containers induce a constantly evolving network topology to which policy must be applied.

Fostering Network Security in a Culture of Agility

By running small, independent services, teams can reduce the bureaucracy and information sharing needed to manage a platform or application. Teams can change the way they interact with the ecosystem, functioning more autonomously, without having to involve a centralized integration team.

Empowering Application Owners

Application owners best understand which interactions are needed for their service to function properly. These teams also typically better understand how information should be classified and which restrictions on data handling should be enforced. This specialization allows data to be isolated by domain and encourages adherence to least-privilege security principles.

The Need for Tiers

While application owners may have the most domain expertise, the reality is that there are still organization-wide policies that require enforcement. Requiring each team to independently incorporate such policies along with their own service-specific policies is impractical for several reasons:

- Teams may not be aware of a given compliance requirement.
- Organization-wide policies may be indistinguishable from service-specific policies.

- Team deployments cannot be autonomous because policies must be audited by a central authority.
- Policy precedence cannot be determined in cases in which policies conflict or are undefined.

As an added layer of complexity, hybrid cloud strategies have further extended the jurisdictions in which policies must be enforced and locality-specific policy overrides may be required. For example, data sovereignty and “safe-harbor” laws may restrict certain types of service interactions.

What is a Policy Tier?

A policy tier is a means of segregating responsibilities for defining network policy. By establishing policy tiers, administrators can separate network policy into logical domains and delegate policy creation to respective domain experts and authorities.

Tier Applications

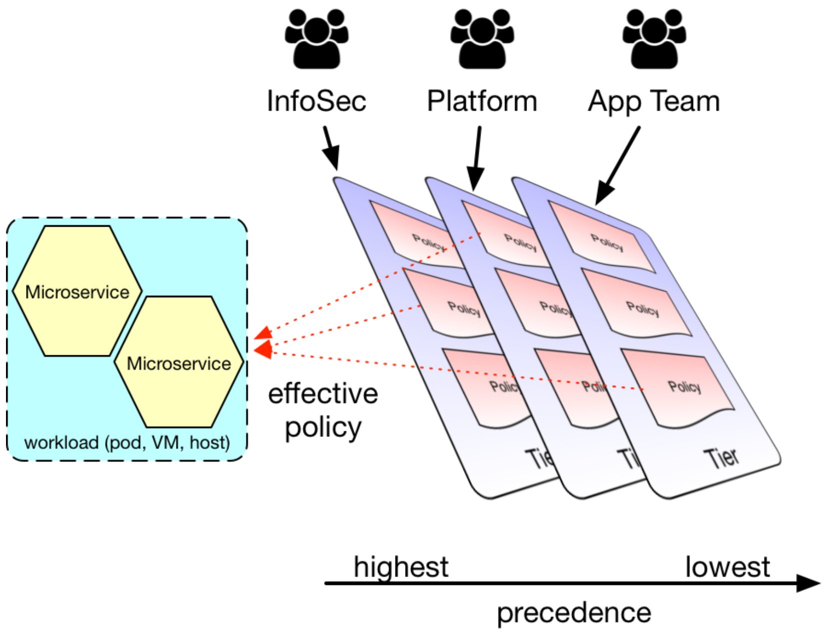
Through policy tiers, service owners can establish policies for their services in one tier, while security, compliance, and operational staff can configure broader policies in separate respective tiers.

Policy tiers are also a mechanism for separating policy concerns into more granular areas. In some cases, temporary overrides and additional policies may be needed to address specific events, such as a network attack. Tiers allow these changes to be applied without editing existing policies.

Tier Precedence

Most importantly, policy tiers are ordered and have a priority that determines policy precedence and thus governs the order of policy application to endpoints. Higher precedence tiers are evaluated and applied first.

Policies can only belong to one tier and must be explicitly associated. Policies that do not specify a tier belong to the default or lowest priority tier. Policies mapped from another framework, such as Kubernetes Network Policies, are also assigned to the default tier.



Tier Cooperation

Tiers must explicitly allow policy evaluation to pass to the next tier in order of precedence; otherwise, policy evaluation will be terminated and the interaction will effectively fail.

Finally, an internal policy tier is simply an ordered collection of policies, as is the standard in Project Calico, the open source networking project managed by Tigera.

Example: Enforcing PCI Compliance

To illustrate how policy tiers are implemented, suppose you have an application governed by the Payment Card Industry Data Security Standard (PCI DSS). To be compliant, components involved in processing financial transactions or handling sensitive financial information must be compliant with the standard. Further, these components must be prohibited from interacting with other components that are not PCI compliant.

Using Labels To Match Policy

In order to apply policies that meet PCI DSS requirements, your deployment only needs to apply a PCI label to PCI-certified components.

A compliance administrator could then create a financial compliance tier and establish policies that restrict any service labeled PCI from interacting with any other service that is missing the PCI label. This would institute a global policy ensuring PCI services only interact with other PCI services.

Authorization Must Be Part of The Solution

In this scenario, only PCI administrators should be able to apply the label signifying PCI compliance. This could be accomplished in multiple ways, such as applying labels in a deployment pipeline, using certificates to assert service identity and so on. Further, the service owner could also implement additional policies to further reduce the exposure of their service, while also adhering to global compliance policies.

Example: NetOps and InfoSec Overrides

Another common use of policy tiers is for network operations (NetOps) or information security (InfoSec) staff to override or change policies at a global level. Policies enforcing global postures are placed in a higher precedence tier and are evaluated before other tiers.

Enforcing DNS Policy

For example, an organization may require all applications to use a set of internal DNS servers for resolving names. Suppose a team creates a container that is configured to use Google's DNS servers instead. To ensure DNS requests can reach the servers, the team also creates a policy enabling both TCP and UDP egress traffic on port 53 to 8.8.8.8 and 8.8.4.4.

If a higher priority policy is established that denies all DNS traffic, except that which is bound for authorized internal servers, DNS traffic would be blocked—regardless of the container-specific policy implemented by the team.

Responding to Security Events

In another use case, an InfoSec team is prompted to respond to a distributed denial-of-service (DDoS) attack affecting multiple services within an organization. In this case, security personnel would want to contain the DDoS threat quickly.

Updating team or service specific policies in such a situation is time-consuming and error-prone. Additionally, once the threat has been neutralized and nominal traffic patterns have been restored, any individual policies that were changed would need to be unwound and restored to their prior state.

It is easier to address these events with a policy tier. Information security staff can mitigate by inserting a new, high-priority policy tier. As a result, every other policy vulnerable to the attack could be overridden. When conditions allow normal operations to resume, the security team could simply remove the tier.

Enterprise Network Policy

Policy tiers enable enterprises to safely delegate policy specification to various teams, which enables agility and encourages granular security policies to be applied to application configuration and deployment. By leveraging this approach along with Tigera Secure, operators and developers can plan and manage their organization's network security posture in a standard and uniform way.

Conclusion

The policy tier construct is a useful way to enforce enterprise policy requirements in a *cloud native* manner. By leveraging this approach, various groups can implement the policies required, without having to return to the "waterfall" security models of the past. Because they represent such a powerful approach, Tigera has employed policy tiers in Tigera Secure.

About Tigera

Tigera delivers solutions for secure application connectivity for the cloud native world. Tigera technology is used by the world's largest enterprises and public cloud providers to power connectivity for application development and deployment and to address the connectivity and security challenges that arise in at-scale production. Tigera Secure meets enterprise needs for zero trust network security, multi-cloud and legacy environment support, organizational control and compliance, and operational simplicity. Secure builds on leading open source projects

Kubernetes, Calico, and Istio, which Tigera engineers help maintain and contribute to as active members of the cloud native community.

tigera.io

email: contact@tigera.io

phone: +1.415.612.9546

Tigera, Inc. 58 Maiden Lane, Fifth Floor, San Francisco CA 94018 USA