Project
Proposal: Enforcing User Defined Privacy Constraint in Service Mesh Data Planes

Ghulam Murtaza
ghulam_murtaza@brown.edu

Amir R. Ilkhechi
ilkhechi@brown.edu

Saim Salman
saim@brown.edu

Abstract

In this project, we propose to add extensions to data plane service meshes (Istio [4], Linkerd [5]) that would automatically generate **policies** to implement user defined policies.

1 Introduction

1.0.1 Service Meshes

The service mesh is a network orchestration layer for micro-services which provides huge management benefits to developers. It abstracts away the complexity and scale involved in managing the micro-services which in turn helps free the developers from tackling a host of security, performance and deployment challenges.

To be more concrete, the most common service mesh implementations (Istio [4], Linkerd [5]) are composed of a control and data plane. The control plane provides the application developers with APIs that help in setting up various policies and providing metrics in turn to the developers. These policies are interpreted into low-level rules which the data plane, a network of Layer 7 proxies, enforces to achieve the high level objectives.

1.0.2 User Policies

In this project, we're restricting ourselves to the **user agreement** or **pop-ups** (asking for permission) a user has to accept to grant the application permission to specific features i.e. Snapchat asks for permission to the Camera when a user signs up.

2 Proposal

2.1 Main Goal

We envision that for all service mesh based web applications, the developer can assign each service (in the application) with specific permissions i.e. the storage service will have the **persistent storage** permission. This list of permissions will be combined with the pop-ups / user agreement forms. For each user, when they allow / dis-allow access to certain services, a respective set of policies will be generated that will ensure that specific services (do / do not) have access to the users data.

Given this interface is implemented, it would by default make the application compliant with GDPR's Article 6 [1], that deals with the consent given by the user to the data processors.

In a simple case, imagine a Mail server that has (among others) an auto-complete and spam filter service. The user can, at any time during their use, opt out of using any of these services. These policies would be then inferred by our tool, that would convert those policies into traffic rules to be deployed at the data plane proxies i.e. Envoy [3] in case of Istio, and they would then in extension make sure that none of the user's traffic gets through unless they explicitly give permission for that interaction to happen.

3 Implementation

To deploy our system, we'll leverage open-sourced, easily accessible service mesh applications (Bookinfo [**?**], Hipster Shop [6], Blueperf [2]) and setup associated permission lists for each application.

Next we'll get Brown undergraduate / graduate students to fill these permission lists to have a list of consent lists for the various applications.

Accordingly, we'll develop an extension that will automatically convert these consent lists to policy rules for the application and get the students to use these application to see how their experience varies.

References

[1] 2018 reform of eu data protection rules.

[2] Blueperf.

[3] Envoy proxy - home.

[4] Istio.

[5] Linkerd.

[6] Google Cloud Platform. Google cloud platform: Microservices demo, Sep 2019.