Abstract

The microservice architecture is a currently emerging pattern in software engineering. Instead of having one huge application, the logic is split into numerous smaller units that fulfill one single purpose. Therefore function calls within the application migrate to remote calls over the network. Remote calls between the services have to provide mutual authentication to secure the system from spoofing. Therefore service-to-service authentication mechanisms are necessary.

The most popular service-to-service authentication mechanisms are self-signed JSON Web Tokens (JWT) and mutual TLS (mTLS). This thesis describes the fundamental concepts and discusses the motivations and challenges of both mechanisms. Furthermore, a project which implements the compared authentication mechanisms is reviewed, and some implementation details are shown. Developers should be able to choos the correct authentication mechanisms for their projects, with the knowledge provided in this thesis.

The comparisons showed that both mechanisms are very efficient and provide the same level of security. Therefore none of the mechanisms is superior for all cases. Self-signed JWTs are the preferred authentication mechanism when nonrepudiation is a requirement, when the application tends to share the user-context, or when the developers require to adapt the authentication mechanism with additional parameters. When none of these requirements apply, mTLS is the preferred approach since it keeps the system as simple as possible.

Kurzfassung

Die Microservice Architektur is ein aufstrebendes Pattern in der Softwareentwicklung. Eine Applikation welche mit der Microservice Architektur aufgebaut ist, besteht aus vielen kleineren Services, die genau einen Zweck erfüllen. Deswegen werden Funktionsaufrufe innerhalb der Applikation zu Netzwerkaufrufen zwischen den Services. Um die Netzwerkkommunikation vor Spoofing zu schützen muss gegenseitige Authentifizierung gewährleistet werden. Hierfür werden Service-zu-Service Authentifizierungsmechanismen verwendet.

Die verbreitetsten Service-zu-Service Authentifizierungsmechanismen sind self-signed JSON Web Tokens (JWT) und mutual TLS (mTLS). Diese Arbeit beschreibt die Ideen und Konzepte, sowie die Motivationen und Herausforderungen dieser Mechanismen. Außerdem wird ein Projekt vorgestellt, welches beide Mechanismen implementiert und Einblicke über die Implementierung dieses Projekts werden gegeben. Mit dem Wissen, aus dieser Arbeit sollen Entwickler in der Lage sein selbst den richtige Authentifizierungsmechanismus für ihre Projekte zu bestimmen.

Die Vergleiche haben gezeigt, dass beide Mechanismen sehr effizient sind und das selbe Maß an Sicherheit liefern. Deswegen ist kein Mechanismus dem anderem gegenüber in jedem Fall überlegen. Self-signed JWTs sind der bevorzugte Authentifizierungsmechanismus, falls Nonrepudiation eine Anforderung ist, wenn die Applikation dazu neigt, den User-Context zu benötigen, oder falls die Entwickler die Authentifizierung selbst mit zusätzlichen Parametern adaptieren möchten. Falls keine dieser Anforderungen zutrifft, ist mTLS der bevorzugte Authentifizierungsmechanismus, da damit das System so einfach wie möglich gehalten wird.