Zentrale und dezentrale Zugriffskontrolle in OAuth2 und OpenID Connect Systemen

OAuth2 als Spezifikation erlaubt es u. a. http-Schnittstellen Token basiert zu sichern, d.h. um Zugriff auf eine durch OAuth2 gesicherte Schnittstelle zu erhalten, ist es notwendig einen sogenannten „access token“ zu besitzen und an die durch OAuth2 gesicherte Schnittstelle zu senden um Zugriff zu erhalten. Ein normaler Ablauf in einem OAuth2 System ist folgendermaßen zu beschreiben: Ein Client erhält dadurch, dass sich ein Nutzer erfolgreich an einem Autorisationsserver authentifiziert einen „access token“ und mit diesem Token kann er auf eine Schnittstelle eines Ressourceservers zugreifen. Der Autorisationsserver ist hierbei das zentrale Element, während die Clients und Ressourceserver eine Vielzahl von heterogenen Applikationen darstellen können. Diese Art der Sicherung ist heutzutage Standard und Unternehmen wie Google und Microsoft nutzen sie in vielen Applikationen.

In der Praxis ist es oftmals notwendig, REST-Schnittstellen rollenbasiert zu sichern, was man standardgemäß als „Role Based Access Control“ (RBAC) bezeichnet. D.h. es kann beispielsweise notwendig sein, manche http-Schnittstellen nur diejenigen Nutzern Zugriff zu gewährleisten, die die Rolle Admin zugeteilt bekommen haben. Grundsätzlich lässt sich dies in jedem System, d.h. in jedem Ressourceserver implementieren („dezentral“), allerdings kann dies schnell bei einer umfangreichen heterogenen Applikationslandschaft mit vielen verschiedenen Programmiersprachen und Frameworks und komplizierten Zugriffsrichtlinien zu Unübersichtlichkeit und einen hohen Wartungsaufwand führen. Aus diesem Grund wurde 2016 das Projekt „Open Policy Agent“ (OPA) entwickelt, dass in einer einheitlichen und leicht verständlichen Programmiersprache, Rego, die Autorisierung, also die Entscheidung ob Zugriff auf eine Schnittstelle erteilt werden darf, von der Applikation entkoppelt („zentral“).

Diese Entkopplung hat allerdings zur Folge, dass der Resourceserver den OPA-Service jedes Mal per http um eine Entscheidung bitten muss, d.h. es besteht das Risiko von gravierenden Performanceeinbußen. WSO2, ein Anbieter eines OAuth2-Autorisierungsservers gibt an, dass ihr Kunde Ebay täglich mehr als eine Milliarde Transaktionen abwickelt (WSO2, kein Datum). Da bei einer hohen Last auf Schnittstellen diese durch horizontale Skalierung entlastet werden, können Performanceeinbußen äußerst kostspielig sein.

Um erstmalig zu untersuchen, inwiefern sich eine Entkopplung von Autorisierungen von der Applikation auf die Performanz auswirkt, wurde prototypisch zwei durch OAuth2 gesicherte Schnittstellen implementiert. In dem einen Testsystem geschieht die Autorisierung in der Applikation und in dem anderen durch einen OPA-Service. Außerdem werden unter dem Gesichtspunkt der Performanz die verschiedenen Deploymentmethoden von OPA analysiert. Um die Performanz unter Last zu testen, wurde als Tool Apache JMeter verwendet und hierbei als zu betrachtende Metrik die sogenannte Round-Trip-Time, also die Zeit die ein Datenpaket braucht um von der Quelle zum Ziel und wieder zurück zu kommen, gewählt. Zudem wurde auf beide Systeme die Sicherheit getestet indem Testfälle entwickelt wurden, mit denen versucht wird auf unerlaubte Weise Zugriff auf die Schnittstelle zu erhalten.

Dabei ist man zu dem Ergebnis gekommen, dass ….

- OPA per Docker-Container = ungefähr halb so schnell

- OPA und Resourceserver in Kubernetes Pod = TODO
- Sicherheit: TODO

Literaturverzeichnis

WSO2. (kein Datum). *eBay Uses 100% Open Source WSO2 ESB to Process More Than 1 Billion Transactions Per Day*. Von wso2.com: https://wso2.com/casestudies/ebay-uses-100-open-source-wso2-esb-to-process-more-than-1-billion-transactions-per-day/ abgerufen