To: Professor Krasso

From: Cory Gilliam

Date: 6 December 2019

Subject: Securing Microservices

In the current atmosphere of web technological advancements, the microservice architecture are the new buzzwords. Having the ability to loosely couple a set of services together in order rapidly build applications contrasts with the old monolithic structure everyone is used to. Having the ability to quickly plug in a new service or replace old ones without extensive code rewrites is appealing. The modular structure of the microservice architecture makes the overall application easer to comprehend and maintain. On the heels of this attractive architecture comes the realization that implementation of security on the application becomes a major challenge.

“There is virtually no situation in software architecture that entirely frees you from security considerations. With microservices, some issues become more distinct and a lot harder. - Marco Troisi”

Microservices must use OAuth 2.0 for access control and authorization handling. The fact that OAuth is supported and used by so many large companies like Facebook, Google, and Twitter, lends credence to the level of trust these companies have placed in its ability to protect from unwanted intrusion. This technology has had lots of capital and development behind its development, almost making it the de facto name in security.

A screenshot of a cell phone

Description automatically generated  
[Pivotal Software, Inc 2013-2015 Creative Commons Attribution-non Commercial license](https://res.infoq.com/presentations/microservices-security-spring-cloud/en/slides/sl11.jpg)

Security must happen in depth, and service security needs to be prioritized. This will help with security by giving your system additional backups. If someone was able to get through one layer of security, they may be stopped by the next. The ability to ensure all the latest security updates are in place could be a difficult and time-consuming task, but the good news is that there is software that can automate this process. This also includes using open source encryption libraries. Open source libraries are community driven, developed, and tested to the point to where it does not make logical or economic sense to try to build a better system.

Firewall with granular controls will also help. The more tightly we can control access to each of the microservices, the more we can fine tune security around their uses. This coupled with building an API gateway is a way to establish a single point of entry will also go a long way in securing the application. By keeping the microservices behind the firewall and out of the public domain attacks will have to go through the API bottleneck.

“Yet microservices are not a magic bullet and implementing a microservices architecture in an organization brings with it some unique challenges. – Sumo Logic”

If we plan for security from the beginning, security will become second nature. Having it be part of the development workflow will go a long way to securing the application and make it maintenance easier.

References

Sumo Logic. (2019, May 14). Improve Security in Microservices Architecture. Retrieved December 6, 2019, from https://www.sumologic.com/insight/microservices-architecture-security/.

Troisi, M. (2019, November 15). 8 best practices for microservices app sec. Retrieved December 6, 2019, from https://techbeacon.com/app-dev-testing/8-best-practices-microservices-app-sec.