To: Teresa Butterfield

From: Adam Donner

Date: 20 August 2019

Subject: Securing Microservices

As an organization we realized that based on our explosive growth we need to move away from our current monolithic application to a microservices design. Microservices takes out aging monolithic application and separates the applications functions in a micro service architecture, each one of these services will perform a task for our current application. An example of this is our current ordering system will be broken up into orders, the basket and the credit card processing, all of these services will of course be communicating with one another to ensure that we do not lose functionality; we will in fact increase scalability, development time and failover. When implementing a change to microservices it is very important to discuss security, as each service talks to one another there needs to be a security layer for that exchange of information. As a development team we have outlined outlined the key features which will be integrated into the microservices architecture.

To make a microservices application usable we will needed to establish how users will access services within it. With this in mind have have established a dedicated servers to function as an API gateway, providing a single point of entry and directing traffic into the different services. By using this technique we can secure all of our services behind a firewall, allowing the API gateway to handle the external requests and then talk to the microservices which will reside behind the firewall. A perfect example of this is the Netflix experience, Susan an API gateway is a perfect way to optimize requests based on the client, for example if they are accessing our site using a mobile device.

We will implement OAuth for user identity and access control. The majority of our services will require some level of access control, OAuth2 is quite possible the industry standard for user authorization. It is a framework that allows users to obtain access to a resource from a resource server, this is done using tokens. These tokens are responsible for access to the resource. The advantage of using OAuth is that we can leverage existing libraries and platform that will accelerate the development phase of this project.

We will also be using a ‘defense in depth’ strategy to prioritize key services. Defense in depth is defined as “information assurance concept in which multiple layers of security controls (defense) are placed through an information technology system.” In other words we have identified what the most sensitive services are and will apply a number of different layers of security to them. The reason for doing so is that potential attackers to our site will need to figure out a way to beet all our our other defenses to the critical layers.

As discussed we will be as we make this change security is a priority and we feel that implementing the above strategies we will be successful in our rollout. The ensure we protect ourselves and our clients we have developed a strategy which I feel will secure the data exchange as well as a strategy to prioritize the key services.

Sincerely,

Adam D. Donner

References

Newizze. (2019, June 14). 7 Best Practices of Microservices Security. Retrieved August 20, 2019, from <https://newizze.com/7-best-practices-of-microservices-security/>

Sumo Logic. Improve Security in Microservices Architecture. Retrieved August 20, 2019, from <https://www.sumologic.com/insight/microservices-architecture-security/>

Troisi, M. (2019, January 22). 8 best practices for microservices app sec. Retrieved August 20, 2019, from https://techbeacon.com/app-dev-testing/8-best-practices-microservices-app-sec