Guide to Securing Microservices

–Sanil Nadkarni,
CISO & VP - SLK GLOBAL SOLUTIONS, sanilnadkarni@yahoo.com

Today's developers are faced with imperceptible challenges of developing code faster and better every day. Building and coding software is always an arduous task especially with monolithic architecture when all codes are all weeded into one solution and are not loosely coupled. Any convulsive change in the code means re-building the entire stack which is often painstaking and time consuming.

Microservices were introduce to simplify these challenges. Microservices solve these inherit lacuna's of monolithic systems by being as modular as possible. With wake of devops and agile software development methodology adopted by most of the software companies Microservices is becoming more popular with the masses.

In the simplest form, microservices help build an application as a suite of small services, each running in its own process and are independently deployable and is woven into one software fabric. These services may be written in different programming languages and may use multiple data storage techniques. Microservices are often connected via APIs, and can leverage many of the same tools and solutions which are used in web services platforms.

Whilst the popularity of microservices begins to see some surge let's look at the security advisory to keep microservices secure -

Authentication and Authorization – Access to microservices should be based on least privilege. Concept of Zero principle should also be adopted for further tightening the reins. Wherever required OAuth authentication should be used. OAuth is an open-standard authorization protocol or framework which outlines how heterogeneous servers and services can safely allow authenticated access to their assets without actually passing and sharing the initial, related, login credential. Wherever possible OAuth use case and existing libraries should be used instead of build your own stack of libraries.

Encryption of sensitive data – Encryption is required for all micro services. The data in motion and data in rest should be encrypted. You could use third party tools or latest cryptographic technology to use the hashing and other algorithm. Use encryption algorithm in accordance with needs of compliance and regulatory requirement such as PCI-DSS ISO, SOX and others certifications.

Network security: - It is highly advisable to implement a cohesive network security controls. Network security hygiene such as anti-virus, HIPS, network NIPS and IDS should be implemented along with automated patching and anti-virus signatory. Further you may improve the security posture by implementing controls such access time windows , identification of devices and geo-locations ip based authorization should also be implemented to have defence in depth.

Testing - The devils are always in the details (lines of code). Testing should be rigorously implemented. Automation of testing should be adopted. Multiple types of testing such as regression/end-to-end/functional testing etc ,should be carried out to ensure that code is well scanned prior to findings its way to the production. The following two types of testing patterns that come to mind to create automated (think JUnit style) tests:

Integrated test of services: these tests are of services compiled by the developers of another services that actually consumes it.

The test suite validated that the service meets the consuming service's requirements.

Component testing of Services: component testing tests the modules in isolation using test doubles for any services that invokes it.

This pattern allows testing by the developers of a single service while not relying on external services

Securing API & Containers – API's are ubiquitous in microservices. Securing API's is of paramount importance. Access to the API's must be allowed in secure manner. Let us look at few ways of securing API's in microservices -

Implementing certificates and validating the parameters should be implemented. To have better security implementing API in distinct tiers.

Screening the code by advance threat detection system before the code is inadvertently activated.

Docker containers are widely adopted in microservices and they come with unique security challenges. To combat these threats one should focus on securing the dockers.

For instance, attackers can compromise an image and get access to application or data files. Further hackers can infect a container with malicious code and use it to attack other containers, the host operating system, or other hosts. Further privilege access of user should be restricted. Discard the SUID flags from your container images. This can further buffet privilege attack escalation.

Never use containers from unofficial resources. The image could be infected. It is also wise to use CoreOS container scanner to validate the registry.

Microservices can be seen as double edge swords while it promises agility, scale and reusability it can also be and by implementing Microservices security intelligently to keep the hackers are bay. ■

Sanil Nadkarni is a executive level management professional with more than 13 + years of core Information Security, Fraud & Risk Management experience with leading Fortune 500 multinational companies. He has authored several articles and news letter in various national and international magazines. He is a speaker and an avid reader.

sanilnadkarni@yahoo.com