YouTube for Software Security? YouTube Videos Provide Pointers for Microservice Security

Md. Shazibul Islam Shamim
Adviser: Dr. Akond Rahman
Department of Computer Science

ABSTRACT

Microservice applications are defined as software applications, which include services that interact with one another but failure of one service does not impact the execution of another. Microservice oriented design has become a popular software application design paradigm among software companies, such as Uber, Netflix, and Amazon as well as small startup companies due to delivery speed, reliability and greater flexibility. However, any insecure coding pattern in the code while developing microservice applications can make the entire system vulnerable to hacker. The goal of the abstract is to help software developers in building secure microservice applications.

BACKGROUND

Previous research in insecure coding pattern for Infrastructure as code scripts[1] inspired us to conduct study on common insecure coding pattern in microservice architecture.

METHODOLOGY

We have conducted a qualitative analysis of 6 YouTube videos[3] on microservice design antipatterns and an empirical study on open source microservice repositories[2]. We have observed insecure coding patterns in those microservice repositories.

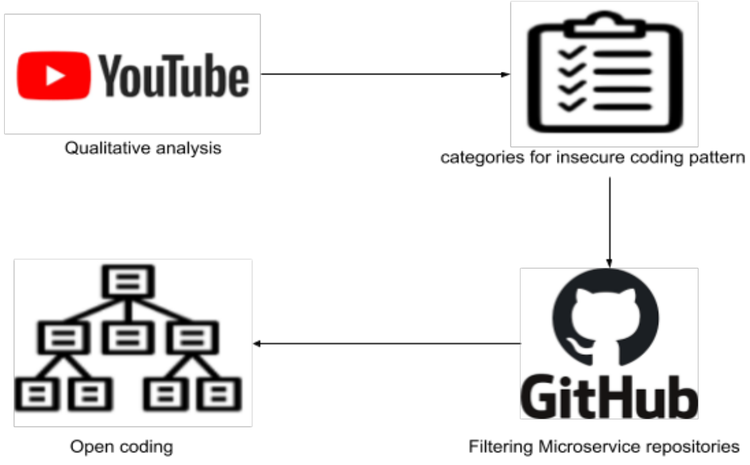

Fig. 1: Diagram of generating insecure coding pattern in microservice architecture .

- We did a qualitative analysis on 6 YouTube videos that demonstrated common insecure coding practices in microservice design
- Initially we defined 17 initial category for potential insecure pattern for microservice coding pattern.
- We filtered GitHub repositories with “Spring-boot” and “microservices” keyword. We observed 30 repositories.
- We did an empirical study on the repositories and searched for initial 17 categories to find evidence of the insecure coding practices.

OBSERVATION

We have observed insecure coding patterns in those microservice repositories. We have found evidence of the following 10 categories each with an associated pattern namely-

1. HTTP without TLS,
2. Authentication vs Authorization,
3. Hard coded secret,
4. Weak encryption algorithm,
5. Use of default ports,
6. Violation of least privilege principle,
7. Insufficient logging,
8. Poor orchestration layer configuration,
9. API service sharing and
10. Distributed deadlock.

CONCLUSION

We advocate for future research that will create a taxonomy of insecure coding patterns so that developers can find and resolve insecure coding patterns during code review.

REFERENCES

[1] Akond Rahman, Chirs Parnin, and Laurie Williams, "The Seven Sins: Security Smells in Infrastructure as Code Scripts", in the International Conference on Software Engineering (ICSE) 2019.
[2] https://github.com/microservices
[3] https://youtube.com/microservices