Software supply chain security in Kubernetes

Degree programme: BSc in Computer Science
Thesis advisor: Prof. Dr. Bruce Nikkel
Expert: Dr. phil. nat. Igor Metz (Glue Software Engineering AG)
Industrial partner: Natron Tech AG, Bern

Attacks on the software supply chain pose a growing threat to the security of software products. Securing and monitoring the supply chain is a very complex and demanding task that requires time and resources. Especially in the cloud environment, there is currently a lack of awareness and guidelines for implementation. This bachelor thesis provides an overview of the topic, the implementation of an application and the enforcement and analysis at runtime.

Jan Fuhrer
IT Security

Introduction

Today, the vast majority of software products are based on third-party code, mostly small Free and Open Source Software (FOSS) components obtained from various sources. In a supply chain attack, it is usually not the code of the application itself, but a directly or indirectly used component that is manipulated to introduce malicious code. The potential for code modification exists at every stage of the typical software supply chain, from source to build, packaging and distribution.

The software supply chain quickly becomes complex as each component depends on others. At each level of abstraction, complexity increases and control decreases. This makes it extremely difficult to ensure the security of the software. Any weakness in the supply chain can therefore undermine confidence in the authenticity of the code being executed, as demonstrated by the recent backdoor attack on the Linux XZ compression library.

Goals

The goal of the thesis is to lower the barrier to entry for developers to implement software supply chain security in their projects, and to provide templates and guidelines for best practices.

Results

This thesis provides a research overview of the current state of software supply chain security frameworks and standards. Based on this, a sample Go application was written on GitHub that implements the Supply-chain Levels for Software Artifacts (SLSA) standard, provides an authenticated Software Bill of Materials (SBOM), and other security and compliance features. Finally, the application was deployed on a Kubernetes cluster with runtime enforcement and component analysis of all running containers in the cluster.
Link to the repository: github.com/janfuhrer/podsalsa

Conclusion

The implementation of software supply chain security is becoming a critical aspect of software development and is one of the most discussed topics in the industry today. More and more applications are following the trend and publishing their artifacts according to the new standards, which is essential for their trust and security. But implementing the standards in a piece of software doesn‘t mean that all of its dependencies have done the same. Supply chain security must be implemented in every dependency of the software, which will be a challenge for the industry.

When it comes to organizational security, an organization needs to have a clear view of the software it uses (and all its dependencies). Component analysis platforms, such as OWASP‘s dependency-track, help to achieve this goal. They provide insight into the tangled and not entirely intuitive ecosystem of open source software. They enable the creation of operational, licensing and security policies to enforce compliance within an organization and to respond quickly to emerging vulnerabilities.

While there is still a lot of work to be done, the direction is clear and we are moving towards a more secure and transparent software supply chain, which will lead to a safer Internet.