# Evaluation and implementation of the SLSA Framework

Degree programme: BSc in Computer Science | Specialisation: IT Security
Thesis advisor: Prof. Hansjürg Wenger
Expert: Dr.Wolfgang Kaltz

Supply chains, like GitLab's CI/CD, are useful for automatically creating and distributing artifacts. However, securing the supply chain is often ignored until it is too late. The SLSA (Supply-chain Levels for Software Artifacts) Framework creates a standard by defining requirements for the supply chains, each requirement defined to increase the supply chain security.

## Introduction

In the last couple of years, many attacks on applications have been reported, where the culprit was not a vulnerability any developer had unknowingly added to the code or a flaw in the applications but in the steps that are taken to create the application from code, the build pipeline. Build pipelines are wonderful helpers to have, as they define the steps to automatically build, test and deploy applications without anyone having to do anything other than pushing code to the build server. However practical a build pipeline is, if a malicious actor, be it an external attacker who gained access to the system or an employee who wants to hurt his employer, has access to the build pipeline, he can do a lot of damage to an application and its users.

## SLSA Framework

To fend off attacks against the pipeline, the SLSA (**S**upply-chain **L**evels for **S**oftware **A**rtifacts, pronounced „salsa") Framework has been created by a collaboration of multiple organizations, among them Google and The Linux Foundation, and is part of the OpenSSF (Open Source Security Foundation). The goal of the SLSA Framework is to ensure that the producer's intent of the software reaches the consumer by creating a standardization for securing the supply chain. This is achieved by posing security requirements to the build pipeline, thus adding resilience and integrity to build artifacts. The requirements defined by the SLSA Framework are separated into three distinct categories; source integrity; on which base the artifacts are generated, dependency integrity; what dependencies are included to produce the artifacts, and build integrity; how the artifacts are generated.

## Goals

This thesis aims to research the SLSA Framework, create a build pipeline that meets the defined requirements, and assess its usefulness, usability, completeness, and integration into GitLab.

## Conclusion

Because the SLSA Framework has existed for not that long, the scope of threat protections in the supply chain is currently not that extensive and only covers some identified threats. Nonetheless, the existing mitigations are implemented and documented well but could be extended by guides. GitLab has only just started to implement the functionality required to provide SLSA compliance. Parts of the requirements defined in SLSA cannot be achieved at all or only achieved by workarounds. This thesis concludes that SLSA already provides a small amount of additional resilience against supply chain threats, but GitLab is not yet ready to make use of it.

Yannick Michael Feller