Statutory declaration

I declare that I have composed the master thesis myself and without use of any other than the cited sources and aids. Sentences or parts of sentences quoted literally are marked as such; other references with regard to the statement and scope are indicated by full details of the publications concerned. The thesis in the same or similar form has not been submitted to any examination body and has not been published. This thesis was not yet, even in part, used in another examination or as a course performance. Furthermore I declare that the submitted written (bound) copies of the master thesis and the version submitted in digital format are consistent with each other in contents.

(Place, Date) (Fabian Burth)

Abstract

The importance of software is constantly growing and so is its complexity. Thus, large-scale enterprise software systems are usually not developed completely from scratch. Commonly required functionality like logging or serialization is often provided by already existing *open source software (OSS)*. Hence, modern software systems are composed of several components. These components usually have individual version updates, vulnerabilities, and licenses. Version updates of a component may affect the compatibility with other components. Vulnerabilities in one specific component may compromise the security of the whole software system. Violating license agreements may lead to litigations due to copyright infringement. Furthermore, one must consider that each component itself can be composed of several other components.

So, maintaining and monitoring large-scale enterprise software systems is an important part of the *Application Lifecycle Management (ALM)* and poses a considerable challenge to software companies. The common way to tackle these risks nowadays is by incorporating *Software Composition Analysis (SCA)* tools into the application development process. These tools analyze applications and retrieve information like vulnerabilities, licenses, and *software bill of materials (SBOM)*. But this approach often still has its flaws. The information extracted by these tools is frequently treated like logs and hence of limited value for future usage. Additionally, in larger companies different development teams often come up with point-to-point solutions of integrating the tools tightly coupled to their development process.

The main objective of this thesis is the design and prototypical implementation of a *Security and Compliance Data Lake (SCDL)*, which provides a standardized way of integrating even multiple different SCA tools into the development process and to store the extracted information. By offering an *Application Programming Interface (API)*, it then should enable consumers to query this information on different levels of aggregation to answer questions that might not even have been known at the time the SCA was performed. A recent and popular example for such a question is "Which components contain log4j?".

This *Security and Compliance Data Lake* will build upon the *Open Component Model (OCM)*, an open standard to describe the SBOM with so called *Component Descriptors*. These *Component Descriptors* also describe how to access sources and resources. They thereby provide an entry point for the execution of SCA tools.

Contents

List of Figures

List of Tables

Acronyms

Chapter 1

Introduction

1.1 Motivation

1.2 Goals

1.3 Environment

1.4 Structure of the Thesis

Chapter 2

Foundations

2.1 Terminology

2.1.1 Vulnerabilities

2.1.2 Package Description Language

2.1.3 OSS Licensing

2.2 Context at SAP Gardener

Builds, Repositories, Gardener Servicelet Pattern (Session mit Uwe)

2.2.1 SAP Gardener

2.2.2 SAP Gardener Deployment Scenario

2.3 Integration into the Application Lifecycle

Chapter 3

State of the Art

3.1 Common Approach of the Industry

3.2 Current Approach at SAP Gardener

3.3 Problems of Existing Approaches

Chapter 4

System Design

4.1 Data Model

4.2 Database

4.3 API

Tool agnostic, how does the api look like to abstract away from different tools

Chapter 5

Implementation

Chapter 6

Results

6.1 Evaluation of the Security and Compliance Data Lake

6.2 Conclusion and Outlook