Master / Bachelor

Infer Security Annotations for Software Models from Requirements / Ableitung von Sicherheitsannotationen für Softwaremodelle mittels Requirements

This thesis can be supervised and written in English as well as in German. For the English version of the announcement see below.

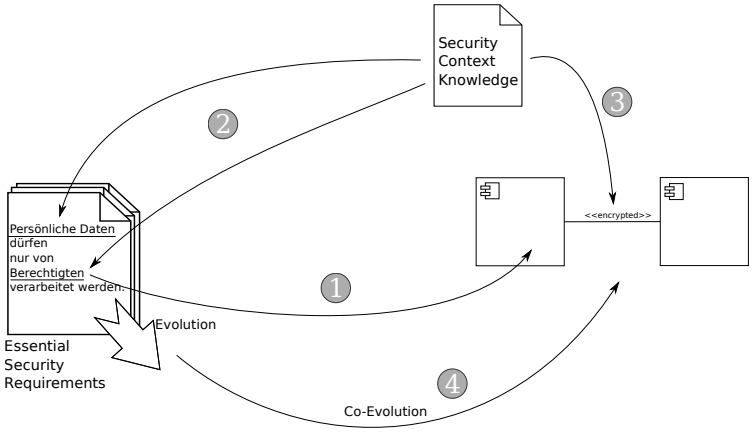

Abbildung 1: Skizze der Ableitung von Annotationen aus ESR und SCK

Motivation

Model-driven software engineering especially is used for compliance to requirements of software systems. Requirements are mostly rather abstract and can come from various sources (e.g. laws, domain-specific regulations, company policies). Using the UML extension UMLsec, one is able to annotate security requirements to models in order to support satisfaction of security requirements throughout the whole development process.

Nowadays, many systems are used over a long timespan (i.e. long-living systems). Many of these are used to provide critical infrastructure, so that livelong compliance to security requirements is of vital significance.

The project *SecVolution* captures and manages knowledge of a specific project and general security knowledge (Security Context Knowledge (SCK)) as well as essential security requirements (ESR) and to make them part of a software model, thus supporting compliance to them during development and even run-time.

If a knowledge evolution comes up, changes to the system model as well as its annotations need to be made.

—

Deutsche Version:

—

Modellgetriebene Softwareentwicklung eignet sich unter anderem dazu, die Einhaltung von Anforderungen an das System sicher zu stellen. Anforderungen (*requirements*) werden auf abstrakter Ebene im Entwicklungsprozess definiert oder sind bereits in Form von externen Quellen vorgegeben. Beispiele für externe Bestandteile von Anforderungsspezifikationen sind Gesetze, domänenspezifische Regularien und firmeninterne Vorgaben. Die UML-Erweiterung UMLsec ermöglicht das Annotieren von Sicherheitsanforderungen an Modelle, sodass beginnend von der Anforderungsspezifikations-Phase über die Modellierung eines Systems bis zur Implementierung durchgängig die Einhaltung von Sicherheitsanforderungen erreicht werden kann.

Software, die in kritischen Bereichen eingesetzt wird, wird oftmals über eine lange Laufzeit betrieben (*langlebiges System*). Langlebige Systeme sind insbesondere mit dem sich verändernden Wissen ihrer Umgebung konfrontiert, wie beispielsweise der Tatsache, dass bestimmte Verschlüsselungsalorithmen, die als sicher gelten, wenn ein System initial entwickelt wird, später unsicher werden können.

Das Projekt *SecVolution* beschäftigt sich damit, Wissen über ein Projekt und allgemeines Sicherheitswissen (Security Context Knowledge (SCK)) sowie generische Sicherheitsanforderungen (Essential Security Requirements (ESR)) zu sammeln und zu verwalten und als Annotationen an das Modell abzubilden, um die Einhaltung dieser Anforderungen bei der Implementierung sicherzustellen.

Ändert sich das Wissen (Evolution), können Änderungen im System-Modell bzw. den Annotationen notwendig werden.

Aufgabenstellung/Ziele

Im Rahmen dieser Arbeit sollen folgende Aspekte der Abb. 1 theoretisch untersucht und die Stichhaltigkeit der Untersuchungen durch prototypische Umsetzung als Tool gezeigt werden:

(1) Aus (angereicherten) ESR sollen (UMLsec-) Annotationen im Modell abgeleitet werden, die diese Anforderungen repräsentieren

(2) Die generischen Begriffe der Essential Security Requirements (ESR) müssen durch konkrete Akteure / Daten, etc. des tatsächlichen Projekts / Domäne, etc. ergänzt werden, damit auf konkrete Modellelemente und dafür erforderliche Sicherheitsanforderungen geschlossen werden kann

(3) Sicherheitsanforderungen, die auf Modellebene vorliegen, sollen durch aktuelles Wissen verfeinert werden. Z. B.: aus der generischen Anforderung ⟨⟨encrypted⟩⟩ für eine verschlüsselte Verbindung einen konkreten Verschlüsselungsalgorithmus und entspr. Parameter ableiten.

(4) ESR können sich ändern. Die Ableitungen, die in Richtung des Modells und Annotationen gemacht wurden, müssen entsprechend adaptiert werden (*Co-Evolution).*

Die Arbeit soll auf die bestehenden Vorarbeiten im Rahmen des Projekts SecVolution sowie dessen Vorarbeiten aufbauen.

Motivation

Model-driven software engineering especially is used for compliance to requirements of software systems. Requirements are mostly rather abstract and can come from various sources (e.g. laws, domain-specific regulations, company policies). Using the UML extension UMLsec, one is able to annotate security requirements to models in order to support satisfaction of security requirements throughout the whole development process.

Nowadays, many systems are used over a long timespan (i.e. long-living systems). Many of these are used to provide critical infrastructure, so that livelong compliance to security requirements is of vital significance.

The project *SecVolution* captures and manages knowledge of a specific project and general security knowledge (Security Context Knowledge (SCK)) as well as essential security requirements (ESR) and to make them part of a software model, thus supporting compliance to them during development and even run-time.

If a knowledge evolution comes up, changes to the system model as well as its annotations need to be made.

Objectives

In this thesis, the following aspects of fig. 1 shall be examined theoretically. The validity of the statements shall then be shown using a software prototype:

(1) Infer UMLsec annotations from (enriched) ESRs

(2) Generic statements of ESRs may have to be refined by concrete actors / data of the actual project / domain, so that derivation of security annotations is made easier or even possible

(3) Security requirements contained in the model have to be refined using current (security) knowledge. For example, then generic security annotation to force data encryption ⟨⟨encrypted⟩⟩ needs to be refined giving a concrete encryption algorithm and appropriate keylength.

(4) ESR can evolve. Derivations that have been made need to be co-evolved.

This thesis relates to the SecVolution project and shall be build by integrating into the existing approaches and tooling.

Organisatorisches

Kontakt:
Dipl.-Inf. Jens Bürger (buerger@uni-koblenz.de)