Karlsruher Institut für Technologie

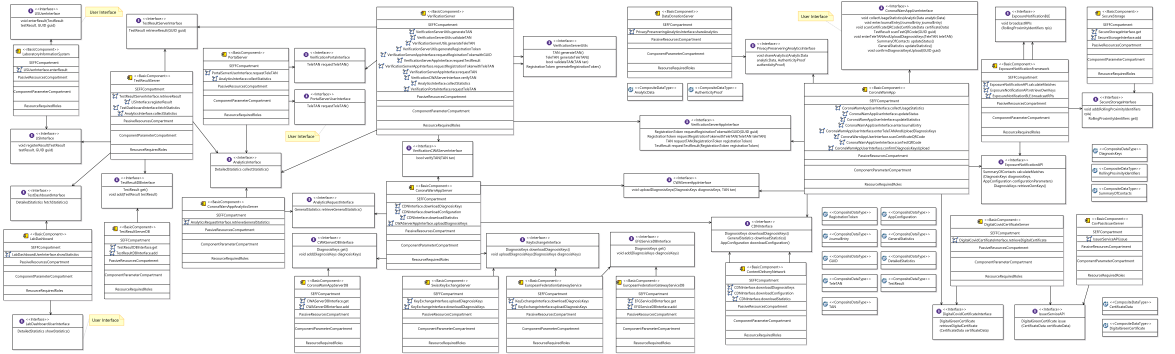
Anforderungsprüfung — ASSUMPTION ANALYZER

Betreuung: Sophie Corallo

Praktikum: Werkzeuge für Agile Modellierung

Tim Bächle | 18. September 2023

Motivation

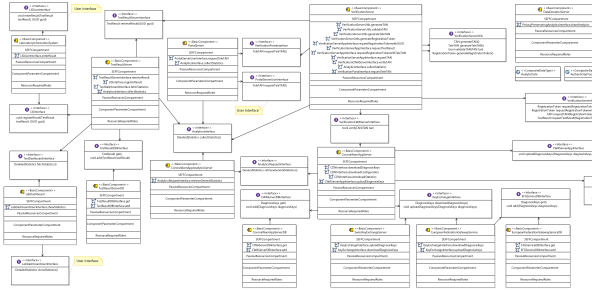


[2]

Motivation

Auf der einen Seite...

- Systeme werden tendenziell immer größer
- Sicherzustellen, dass große Systeme alle Anforderungen erfüllen ist schwierig
- Besonders Sicherheit ist ein wichtiger Aspekt
  - Regulatorisch
  - Interesse der Öffentlichkeit



[2]

Motivation

Auf der einen Seite...

- Systeme werden tendenziell immer größer
- Sicherzustellen, dass große Systeme alle Anforderungen erfüllen ist schwierig
- Besonders Sicherheit ist ein wichtiger Aspekt
  - Regulatorisch
  - Interesse der Öffentlichkeit

Auf der anderen Seite...

- Schon die Architektur kann mit Sicherheitsanalysen analysiert werden
- Verschiedene Analysen betrachten unterschiedliche Sicherheitsaspekte
  - ABUNAI [1] betrachtet z.B. die Vetraulichkeit (Confidentiality)
- Die Nutzung mehrerer Analysen kann aufwendig sein
  - Installation, Nutzung, . . .

Motivation

Auf der einen Seite...

- Systeme werden tendenziell immer größer
- Sicherzustellen, dass große Systeme alle Anforderungen erfüllen ist schwierig
- Besonders Sicherheit ist ein wichtiger Aspekt
  - Regulatorisch
  - Interesse der Öffentlichkeit

Auf der anderen Seite...

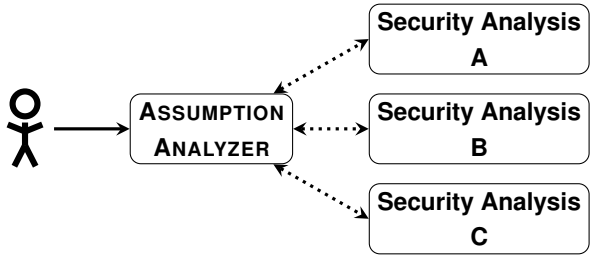
- Schon die Architektur kann mit Sicherheitsanalysen analysiert werden
- Verschiedene Analysen betrachten unterschiedliche Sicherheitsaspekte
  - ABUNAI [1] betrachtet z.B. die Vetraulichkeit (Confidentiality)
- Die Nutzung mehrerer Analysen kann aufwendig sein
  - Installation, Nutzung, . . .

$\Rightarrow$ Lässt sich das Auswerten mehrerer Sicherheitsanalysen einfacher gestalten?

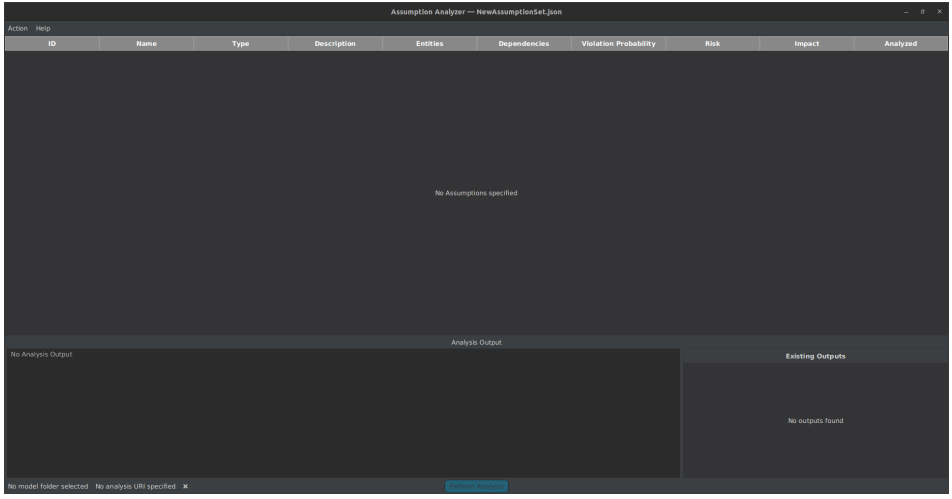
ASSUMPTION ANALYZER — Idee

Übergeordnete Ziele:

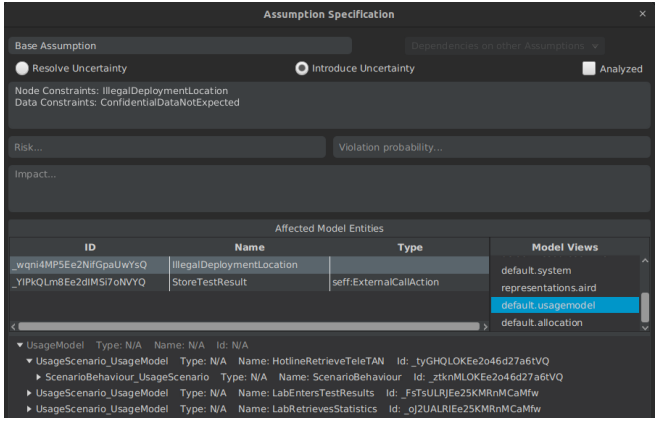
- Eine Anwendung
- Unterstützung diverser Sicherheitsanalysen
- Einfache Handhabung



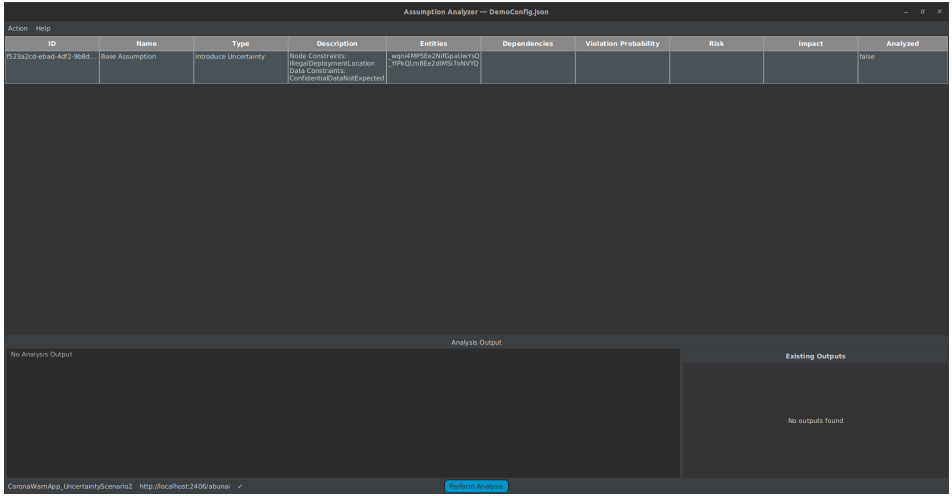
ASSUMPTION ANALYZER — Anwendung



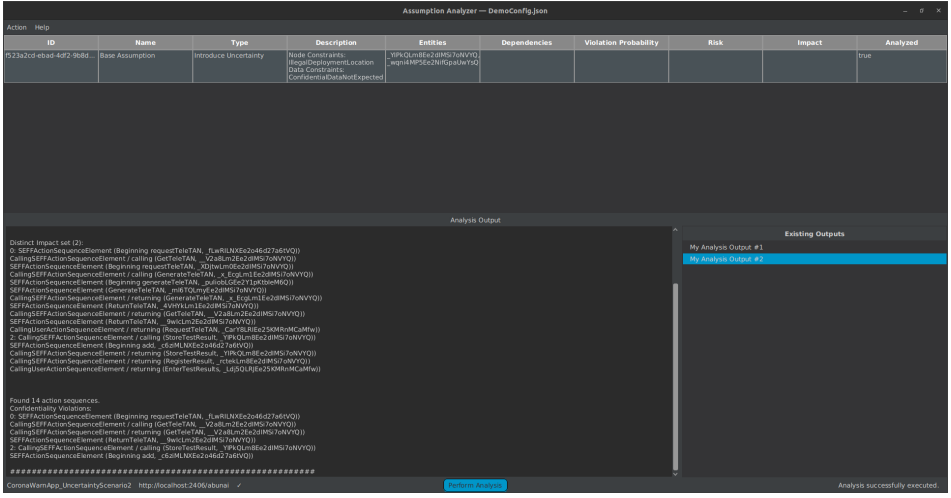
Assumption Analyzer — Anwendung



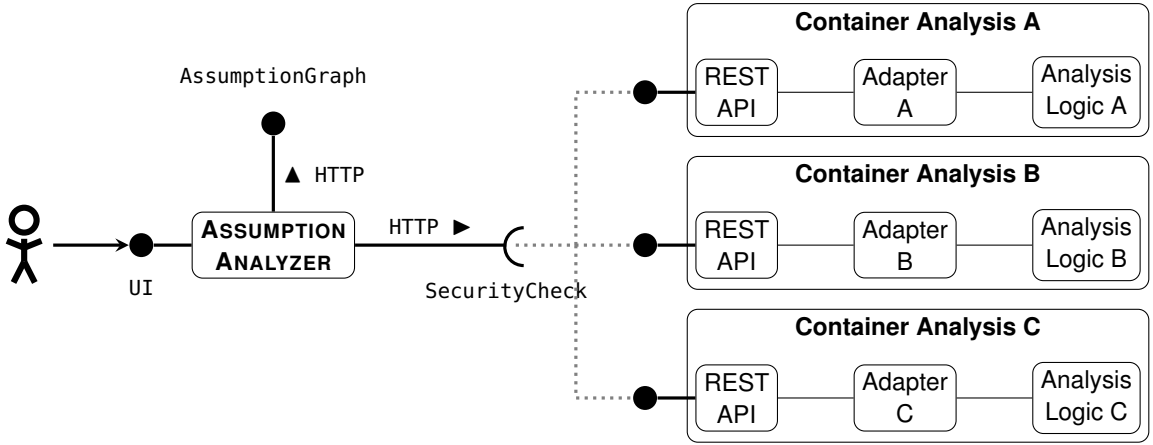
ASSUMPTION ANALYZER — Anwendung



Assumption Analyzer — Anwendung



ASSUMPTION ANALYZER — Architektur



Anbindung an Sicherheitsanalysen

Direkte Integration

- Integriere UI direkt in die einzelnen Analysen

- ⊕ Einfach
- ⊖ Schlechte Wartbarkeit
- ⊖ Eine Anwendung pro Analyse

Integration über CLI

- Statte die Analysen mit einem standardisierten CLI aus

- ⊕ Eine Anwendung für viele Analysen
- ⊖ Standardisierte CLI komplex
- ⊖ Hänge von absolutem Pfad der Analyse ab

Analysen als Microservices

- Statte die Analysen mit einer REST-Schnittstelle aus

- ⊕ Eine Anwendung für viele Analysen
- ⊕ Vorteile einer Microservice Architektur
- ⊕ Containerisierung bietet sich an

Anbindung an Sicherheitsanalysen

Direkte Integration

- Integriere UI direkt in die einzelnen Analysen

- ⊕ Einfach
- ⊖ Schlechte Wartbarkeit
- ⊖ Eine Anwendung pro Analyse ↯

Integration über CLI

- Statte die Analysen mit einem standardisierten CLI aus

- ⊕ Eine Anwendung für viele Analysen
- ⊖ Standardisierte CLI komplex
- ⊖ Hänge von absolutem Pfad der Analyse ab

Analysen als Microservices

- Statte die Analysen mit einer REST-Schnittstelle aus

- ⊕ Eine Anwendung für viele Analysen
- ⊕ Vorteile einer Microservice Architektur
- ⊕ Containerisierung bietet sich an

Umgang mit Nutzerdaten — Ungespeicherte Änderungen

Boolean Flag

- Setzte Boolean Flag bei Änderung, Rücksetzen bei Speichervorgang
- ⊕ Einfach umzusetzen
- ⊖ Fehleranfällig: Evolution & Wartung
- ⊖ Muss bei Änderungen der Logik der Applikation angepasst werden

Vergleiche Serialisierung

- Vergleiche die Strings der JSON-Serialisierung
- ⊕ Sehr einfach umzusetzen
- ⊕ Unabhängig von Logik der Applikation
- ⊖ Fehleranfällig: Ungespeicherte Änderungen bei anderer Serialisierung?

Vergleiche Konfiguration

- Vergleiche Konfiguration der Anwendung mit der letzten gespeicherten
- ⊕ Robust
- ⊖ Benötige Mechanismus zum Klonen
- ⊖ Muss bei Änderungen der Konfiguration der Applikation angepasst werden

Umgang mit Nutzerdaten — Ungespeicherte Änderungen

Boolean Flag

- Setzte Boolean Flag bei Änderung, Rücksetzen bei Speichervorgang
- ⊕ Einfach umzusetzen
- ⊖ Fehleranfällig: Evolution & Wartung
- ⊖ Muss bei Änderungen der Logik der Applikation angepasst werden

Vergleiche Serialisierung

- Vergleiche die Strings der JSON-Serialisierung
- ⊕ Sehr einfach umzusetzen
- ⊕ Unabhängig von Logik der Applikation
- ⊖ Fehleranfällig: Ungespeicherte Änderungen bei anderer Serialisierung?

Vergleiche Konfiguration

- Vergleiche Konfiguration der Anwendung mit der letzten gespeicherten
- ⊕ Robust
- ⊖ Benötige Mechanismus zum Klonen
- ⊖ Muss bei Änderungen der Konfiguration der Applikation angepasst werden

Ausblick

Abunai Microservice

- Erwartet in aktueller Version Constaints in festem Format
- Potenzial, Constraints per NLP aus der Beschreibung der Assumption abzuleiten

Ausblick

Abunai Microservice

- Erwartet in aktueller Version Constaints in festem Format
- Potenzial, Constraints per NLP aus der Beschreibung der Assumption abzuleiten

Komplexe Abhängigkeiten

- Bisher nur direkte Abhängigkeiten zwischen Assumptions
- Potenzial, auch komplexe Abhängigkeiten (Verknüpfung mit `OR`, `AND`, etc.) einzuführen

Ausblick

Abunai Microservice

- Erwartet in aktueller Version Constaints in festem Format
- Potenzial, Constraints per NLP aus der Beschreibung der Assumption abzuleiten

Komplexe Abhängigkeiten

- Bisher nur direkte Abhängigkeiten zwischen Assumptions
- Potenzial, auch komplexe Abhängigkeiten (Verknüpfung mit `OR`, `AND`, etc.) einzuführen

Weitere Analysen

- Bisher nur Abunai unterstützt
- Weitere PCM-basierte Analysen lassen sich leicht hinzufügen

Vielen Dank für eure Aufmerksamkeit!

Literatur

[1] Sebastian Hahner. *Architecture-Based Uncertainty-Aware Confidentiality Analysis*. 2023. URL: https://github.com/abunai-dev (besucht am 23. 08. 2023).

[2] Sebastian Hahner. *Case Study: Corona Warn App*. 2023. URL: https://github.com/abunai-dev/CaseStudy-CoronaWarnApp (besucht am 15. 09. 2023).