

Anforderungsprüfung — ASSUMPTION ANALYZER

Betreuung: Sophie Corallo

Praktikum: Werkzeuge für Agile Modellierung

Tim Bächle | 18. September 2023

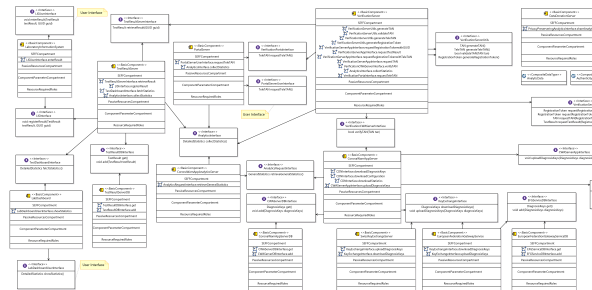




Motivation

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]

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 Vertraulichkeit (Confidentiality)
- Die Nutzung mehrerer Analysen kann aufwendig sein
 - Installation, Nutzung, . . .

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...

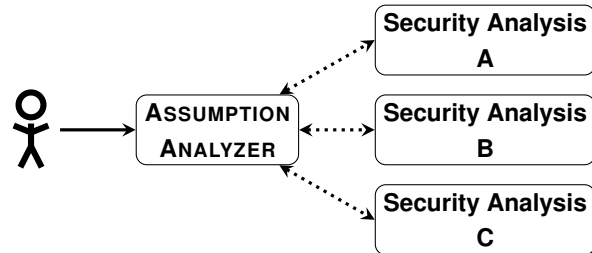
- Schon die Architektur kann mit Sicherheitsanalysen analysiert werden
- Verschiedene Analysen betrachten unterschiedliche Sicherheitsaspekte
 - ABUNAI [1] betrachtet z.B. die Vertraulichkeit (Confidentiality)
- Die Nutzung mehrerer Analysen kann aufwendig sein
 - Installation, Nutzung, ...

⇒ Lässt sich das Auswerten mehrerer Sicherheitsanalysen einfacher gestalten?

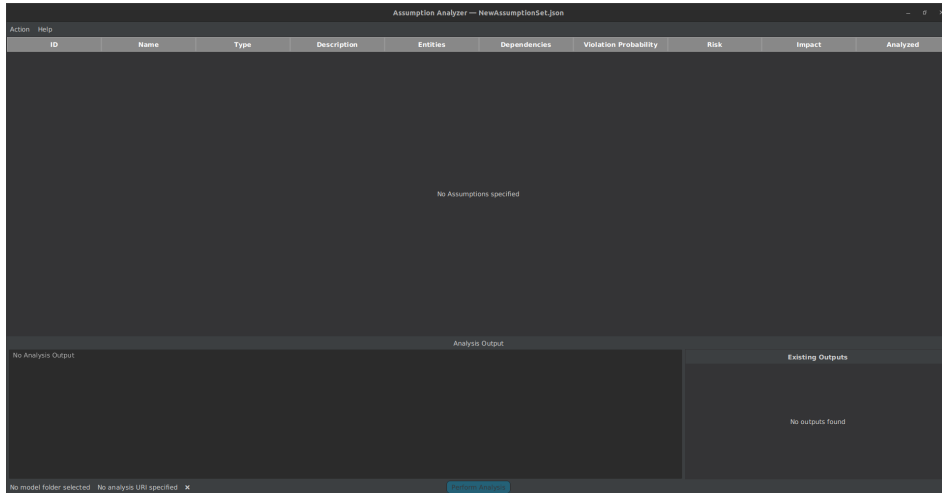
ASSUMPTION ANALYZER — Idee

Übergeordnete Ziele:

- Eine Anwendung
- Unterstützung diverser Sicherheitsanalysen
- Einfache Handhabung



ASSUMPTION ANALYZER — Anwendung



ASSUMPTION ANALYZER — Anwendung

Assumption Specification

Base Assumption

Dependencies on other Assumptions

☐ Resolve Uncertainty
☒ Introduce Uncertainty
☐ Analyzed

Node Constraints: IllegalDeploymentLocation
Data Constraints: ConfidentialDataNotExpected

Risk...

Violation probability...

Impact...

Affected Model Entities

ID	Name	Type	Model Views
wqni4MP5Ee2NifGpaUwYsQ	IllegalDeploymentLocation		default.system
YIPkQLm8Ee2diMSi7oNVYQ	StoreTestResult	seff:ExternalCallAction	representations.aird
			default.usagemodel
			default.allocation

UsageModel Type: N/A Name: N/A Id: N/A

UsageScenario_UsageModel Type: N/A Name: HotlineRetrieveTeleTAN Id: _tyGHQLOKEe2o46d27a6tVQ

ScenarioBehaviour_UsageScenario Type: N/A Name: ScenarioBehaviour Id: _ztknMLOKEe2o46d27a6tVQ

UsageScenario_UsageModel Type: N/A Name: LabEntersTestResults Id: _FsTsULRJJe25KMRnMCaMfw

UsageScenario_UsageModel Type: N/A Name: LabRetrievesStatistics Id: _oj2UALRIEe25KMRnMCaMfw

ASSUMPTION ANALYZER — Anwendung

Assumption Analyzer — DemoConfig.json

ID	Name	Type	Description	Entities	Dependencies	Violation Probability	Risk	Impact	Analyzed
f523a2cd-ebad-4d72-9b8d...	Base Assumption	Introduce Uncertainty	Node Constraints: IllegalDeploymentLocation Data Constraints: ConfidentialDataNotExpected	wqn4MP5Ee2NirGpaUwYQ _YpKOLm8EeZdlMSI7oNVYQ					false

Analysis Output

No Analysis Output

Existing Outputs

No outputs found

CoronaWarnApp_UncertaintyScenario2 http://localhost:2406/abunai ✓

Perform Analysis

ASSUMPTION ANALYZER — Anwendung

Assumption Analyzer — DemoConfig.json

Action Help

ID	Name	Type	Description	Entities	Dependencies	Violation Probability	Risk	Impact	Analyzed
IS23a2cd-ebad-4d12-9b8d...	Base Assumption	Introduce Uncertainty	Node Constraints: [logaDeploymentLocation Data Constraints: ConfidentialDataNotExpected	_YpKQLm8Ee2dIM5i7oNVYQ _wqn4MP5Ee2NiIfGaUwYQ					true

Analysis Output

Distinct Impact set (2):

```

0: SEFFActionSequenceElement (Beginning requestTeleTAN, _flwRILNXEo2o46d27a6tvQ))
CallingSEFFActionSequenceElement / calling (GetTeleTAN, _V2a8Lm2Ee2dIM5i7oNVYQ))
SEFFActionSequenceElement (Beginning requestTeleTAN, _XDjWLM0Ee2dIM5i7oNVYQ))
CallingSEFFActionSequenceElement / calling (GenerateTeleTAN, _x_Ecp.m1Ee2dIM5i7oNVYQ))
SEFFActionSequenceElement (Beginning generateTeleTAN, _pulioblGee2Y1pKtsie6MQ))
SEFFActionSequenceElement (GenerateTeleTAN, _mB1QQLmYle2dIM5i7oNVYQ))
CallingSEFFActionSequenceElement / returning (GenerateTeleTAN, _s_Ecp.m1Ee2dIM5i7oNVYQ))
SEFFActionSequenceElement (ReturnTeleTAN, _4VhYKLM1Ee2dIM5i7oNVYQ))
CallingSEFFActionSequenceElement / returning (GetTeleTAN, _V2a8Lm2Ee2dIM5i7oNVYQ))
SEFFActionSequenceElement (ReturnTeleTAN, _9wicLm2Ee2dIM5i7oNVYQ))
CallingUserActionSequenceElement / returning (RequestTeleTAN, _CsY8LR1Ee25K0MnMcMfw))
2: CallingSEFFActionSequenceElement / calling (StoreTestResult, _YpKQLm8Ee2dIM5i7oNVYQ))
SEFFActionSequenceElement (Beginning add, _c6oMLNXEe2o46d27a6tvQ))
CallingSEFFActionSequenceElement / returning (StoreTestResult, _YpKQLm8Ee2dIM5i7oNVYQ))
CallingSEFFActionSequenceElement / returning (RegisterResult, _rctekLm8Ee2dIM5i7oNVYQ))
CallingUserActionSequenceElement / returning (EnterTestResult, _ldjSQLJEe25K0MnMcMfw))

```

Found 14 action sequences.
Confidentiality Violations:

```

0: SEFFActionSequenceElement (Beginning requestTeleTAN, _flwRILNXEo2o46d27a6tvQ))
CallingSEFFActionSequenceElement / calling (GetTeleTAN, _V2a8Lm2Ee2dIM5i7oNVYQ))
CallingSEFFActionSequenceElement / returning (GetTeleTAN, _V2a8Lm2Ee2dIM5i7oNVYQ))
SEFFActionSequenceElement (ReturnTeleTAN, _9wicLm2Ee2dIM5i7oNVYQ))
2: CallingSEFFActionSequenceElement / calling (StoreTestResult, _YpKQLm8Ee2dIM5i7oNVYQ))
SEFFActionSequenceElement (Beginning add, _c6oMLNXEe2o46d27a6tvQ))

```

CoronaWarnApp_UncertaintyScenario2 http://localhost:2406/abunai ✓

Perform Analysis

Existing Outputs

My Analysis Output #1

My Analysis Output #2

Analysis successfully executed.

Motivation

5/10

18.9.2023

Assumption Analyzer

Tim Bächle: Anforderungsprüfung

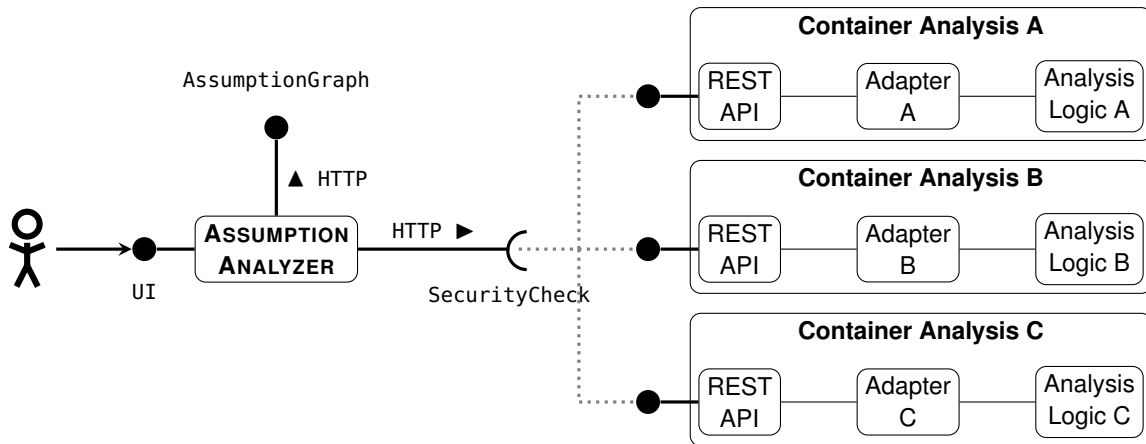
Architektur

Ausgewählte Designentscheidungen

Ausblick

KASTEL – Institute of Information Security and Dependability

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

Abunai Microservice

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

Abunai Microservice

- Erwartet in aktueller Version Constraints 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

Abunai Microservice

- Erwartet in aktueller Version Constraints 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!

- [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).