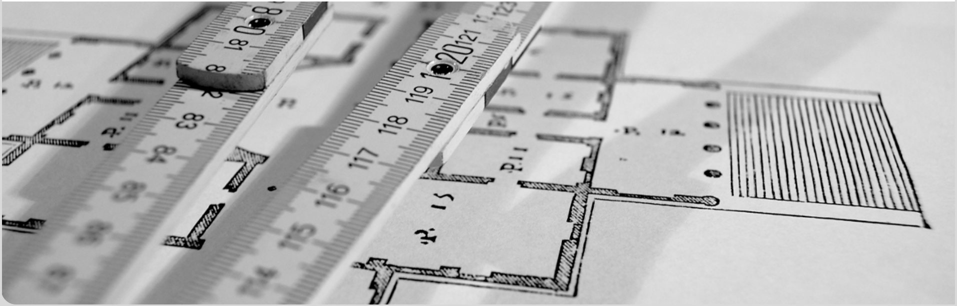
Access Control Verification in Software Systems

Bachelor's thesis

Reviewer: Prof. Dr. Ralf H. Reussner, Jun.-Prof. Dr.-Ing. Anne Koziolek

Julian Hinrichs | October 1, 2018

CHAIR FOR SOFTWARE DESIGN AND QUALITY



Motivation

- Architectural security analysis
  - Save resources
  - Adapt the system model in an early design stage.
  - Avoid inconsistency between the security documentation and the system model.
- Different approaches: Data-based privacy analysis(DPBA) (Seifermann 2016), UMLsec (Jürjens 2002), etc
- The evaluation of DBPA approaches is not carried out formally, but through case studies.
- It is not trivial to create case studies.
- Goal: to create case studies to evaluate privacy defined by access rights.

Motivation

- Architectual security analysis
  - Save resources
  - Adapt the system model in an early design stage.
  - Avoid inconsistency between the security documentation and the system model.
- Different approaches: Data-based privacy analysis(DPBA) (Seifermann 2016), UMLsec (Jürjens 2002), etc
- The evaluation of DBPA approaches is not carried out formally, but through case studies.
- It is not trivial to create case studies.
- Goal: to create case studies to evaluate privacy defined by access rights.

Motivation

- Architectural security analysis
  - ~~Save resources~~
  - Adapt the system model in an early design stage.
  - Avoid inconsistency between the security documentation and the system model.
- Different approaches: Data-based privacy analysis(DPBA) (Seifermann 2016), UMLsec (Jürjens 2002), etc
- The evaluation of DBPA approaches is not carried out formally, but through case studies.
- It is not trivial to create case studies.
- Goal: to create case studies to evaluate privacy defined by access rights.

Motivation

- Architectural security analysis
  - Save resources
  - Adapt the system model in an early design stage.
  - Avoid inconsistency between the security documentation and the system model.
- Different approaches: Data-based privacy analysis(DPBA) (Seifermann 2016), UMLsec (Jürjens 2002), etc
- The evaluation of DBPA approaches is not carried out formally, but through case studies.
- It is not trivial to create case studies.
- Goal: to create case studies to evaluate privacy defined by access rights.

Motivation

- Architectural security analysis
  - Save resources
  - Adapt the system model in an early design stage.
  - Avoid inconsistency between the security documentation and the system model.
- Different approaches: Data-based privacy analysis(DPBA) (Seifermann 2016), UMLsec (Jürjens 2002), etc
- The evaluation of DBPA approaches is not carried out formally, but through case studies.
- It is not trivial to create case studies.
- Goal: ~~to~~ create case studies to evaluate privacy defined by access rights.

Related work

- Case studies are already used in software engineering (Runeson and Höst 2009).
- **Similarities**
  - Process for creating a case study, goal of the case study, etc.
- **Differences**
  - Usually the solutions to a problem are investigated, we examine the data processing in a concrete system
- Requirements for privacy
  - The problem statement non-influence (Oheimb 2004) defines requirements for privacy.
- Evered and Bögeholz 2004 is a relatebale source
  - Measurement for good access rights
  - Case study example for a much smaller scope.

Related work

- Case studies are already used in software engineering (Runeson and Höst 2009).
- **Similarities**
  - Process for creating a case study, goal of the case study, etc.
- **Differences**
  - Usually the solutions to a problem are investigated, we examine the data processing in a concrete system
- Requirements for privacy
  - The problem statement non-influence (Oheimb 2004) defines requirements for privacy.
- Evered and Bögeholz 2004 is a relatebale source
  - Measurement for good access rights
  - Case study example for a much smaller scope.

Related work

- Case studies are already used in software engineering (Runeson and Höst 2009).
- **Similarities**
  - Process for creating a case study, goal of the case study, etc.
- **Differences**
  - Usually the solutions to a problem are investigated, we examine the data processing in a concrete system
- Requirements for privacy
  - The problem statement non-influence (Oheimb 2004) defines requirements for privacy.

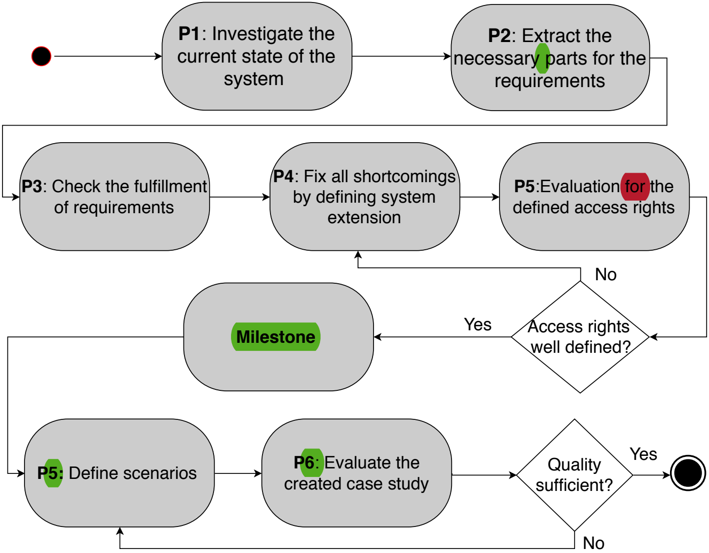
- Evered and Bögeholz 2004 is a relatebale source
  - Measurement for good access rights
  - Case study example for a much smaller scope.

Related work

- Case studies are already used in software engineering (Runeson and Höst 2009).
- **Similarities**
  - Process for creating a case study, goal of the case study, etc.
- **Differences**
  - Usually the solutions to a problem are investigated, we examine the data processing in a concrete system
- Requirements for privacy
  - The problem statement non-influence (Oheimb 2004) defines requirements for privacy.

- Evered and Bögeholz 2004 is a relatebale source
  - Measurement for good access rights
  - Case study example for a much smaller scope.

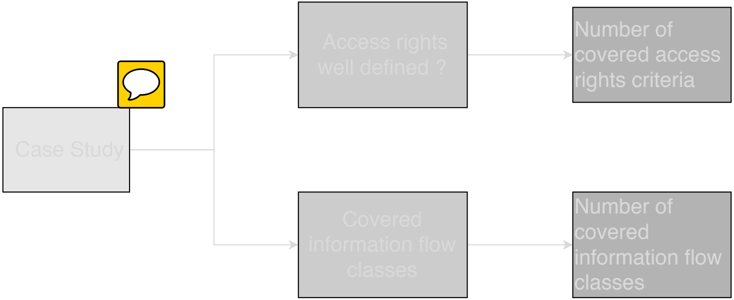
Procedure Overview



P1: Investigate the current state of the system
P2: Extract the necessary parts for the requirements
P3: Check the fulfillment of requirements
P4: Fix all shortcomings by defining system extension
P5:Evaluation for the defined access rights
No
Access rights well defined?
Yes
Milestone
P5: Define scenarios
P6: Evaluate the created case study
Quality sufficient?
Yes
No

Evaluation for the case study

- Evaluation of the case study is split in two parts:
  - Evaluation of the access rights right before the milestone to confirm it is reached.
  - Evaluation of the defined scenarios to decide if the procedure is concluded.



Case Study

Access rights well defined ?

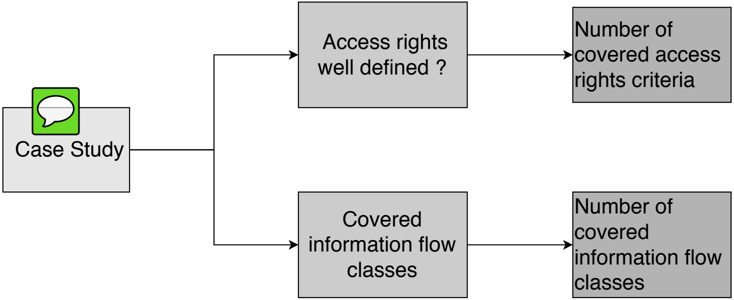
Number of covered access rights criteria

Covered information flow classes

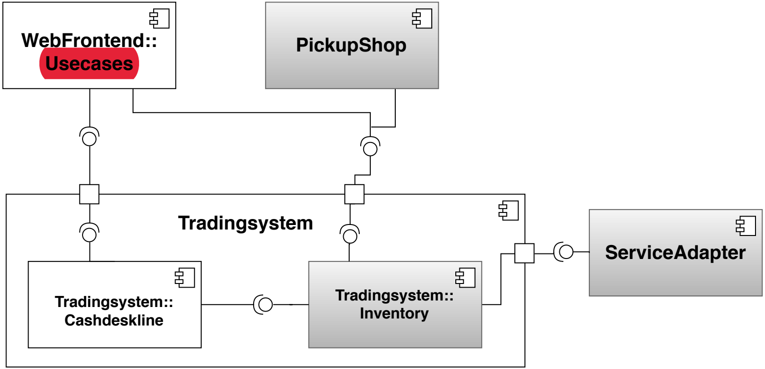
Number of covered information flow classes

Evaluation for the case study

- Evaluation of the case study is split in two parts:
  - Evaluation of the access rights right before the milestone to confirm it is reached.
  - Evaluation of the defined scenarios to decide if the procedure is concluded.


Case Study
Access rights well defined ?
Number of covered access rights criteria
Covered information flow classes
Number of covered information flow classes

P1: Investigate the current state of CoCoME



Requirements for privacy-considering case study

Requirements	
R1	component based system
R2	Definition of use cases
R3	Security relevant data
R4	Definition of user roles
R5	Definition of access rights
R6	Definition of the type of data processing in the components

Procedure P2 -P4: Requirements R1-R4

- ✓: documentation, ▲: defined, ●: generated
- R1: Component based system ✓
- R2: Use cases ✓
  - 13 use cases are defined in the documentation
- R3: Security relevant data ▲
  - Four different classes for the data in CoCoME
  - The security relevance for each class was measured according to(Breier 2014)
  - Account data: security relevant
  - Customer data: security relevant
  - System data: security relevant
  - P& S data: security relevant in composition with one of the other classes.
- R4: User roles ✓— ▲
  - 5 roles are defined in the documentation
  - some roles needed some refinement.

Procedure P2 -P4: Requirements R1-R4

- ✓: documentation, ▲: defined, ●: generated
- R1: Component based system ✓
- R2: Use cases ✓
  - 13 use cases are defined in the documentation
- R3: Security relevant data ▲
  - Four different classes for the data in CoCoME
  - The security relevance for each class was measured according to(Breier 2014)
  - Account data: security relevant
  - Customer data: security relevant
  - System data: security relevant
  - P& S data: security relevant in composition with one of the other classes.
- R4: User roles ✓— ▲
  - 5 roles are defined in the documentation
  - some roles needed some refinement.

Procedure P2 -P4: Requirements R1-R4

- ✓: documentation, ▲: defined, ●: generated
- R1: Component based system ✓
- R2: Use cases ✓
  - 13 use cases are defined in the documentation
- R3: Security relevant data ▲
  - Four different classes for the data in CoCoME
  - The security relevance for each class was measured according to(Breier 2014)
  - Account data: security relevant
  - Customer data: security relevant
  - System data: security relevant
  - P& S data: security relevant in composition with one of the other classes.
- R4: User roles ✓— ▲
  - 5 roles are defined in the documentation
  - some roles needed some refinement.

Procedure P2 -P4: Requirements R1-R4

- ✓: documentation, ▲: defined, ●: generated
- R1: Component based system ✓
- R2: Use cases ✓
  - 13 use cases are defined in the documentation
- R3: Security relevant data ▲
  - Four different classes for the data in CoCoME
  - The security relevance for each class was measured according to(Breier 2014)
  - Account data: security relevant
  - Customer data: security relevant
  - System data: security relevant
  - P& S data: security relevant in composition with one of the other classes.
- R4: User roles ✓— ▲
  - 5 roles are defined in the documentation
  - some roles needed some refinement.

Procedure P2-P4: access rights ●

- Derived from the previous requirements R1-R4
- Finer grained, high level form derived from (Evered and Bögeholz 2004)
- Access control matrix (ACM)

ACM	Webfrontend		TS:Inventory	
StockManager	customer data	4	Customer data	4
	account data	3	Account data	3
	p&s data	2	P& S data	2
	system data	4	System data	4

Table: Level 1: **fullAccess**, Level 2: **AccessToUsedData**, Level 3: **AccessToOwnData**, Level 4: **default**

Procedure P2-P4: access rights ●

- Derived from the previous requirements R1-R4
- Finer grained, high level form derived from (Evered and Bögeholz 2004)
- Access control matrix (ACM)

ACM	Webfrontend		TS:Inventory	
StockManager	customer data	4	Customer data	4
	account data	3	Account data	3
	p&s data	2	P& S data	2
	system data	4	System data	4

Table: Level 1: **fullAccess**, Level 2: **AccessToUsedData**, Level 3: **AccessToOwnData**, Level 4: **default**

Procedure P2-P4: Types of data processing in the system ●

- We identified four categories of data processing in CoCoME.
  - Transmission of data
  - alternation of data
  - relational algebra
  - I/O
- Operations matrix(OpM)

OpM	customer	account	P& S	system
Webfrontend	transmit	transmit	I/O, transmit	n/a

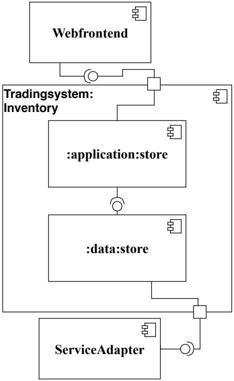
Procedure P2-P4: Types of data processing in the system ●

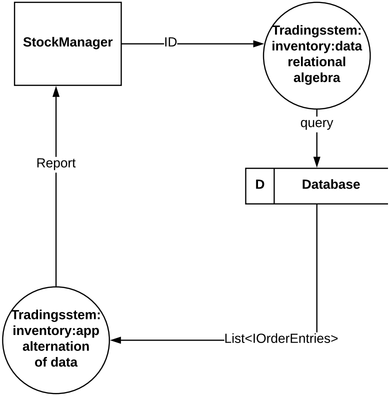
- We identified four categories of data processing in CoCoME.
  - Transmission of data
  - alternation of data
  - relational algebra
  - I/O
- Operations matrix(OpM)

OpM	customer	account	P& S	system
Webfrontend	transmit	transmit	I/O, transmit	n/a

Procedure P5: Definition of a Scenario

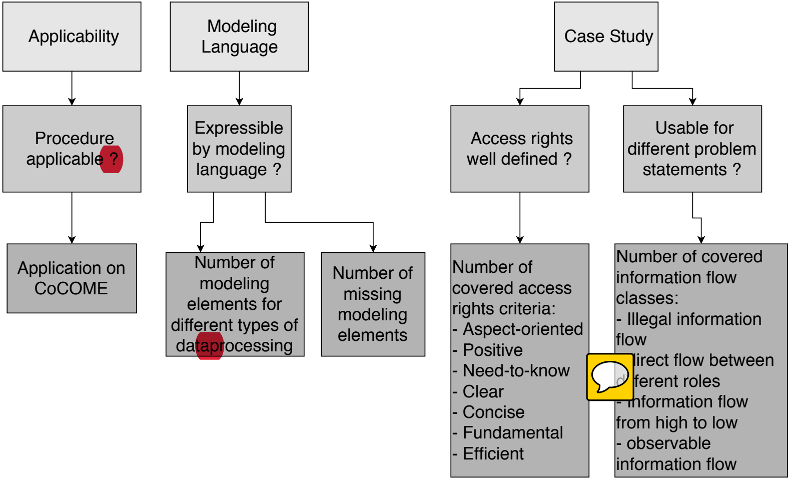
- Scenario: StockManager requests a report for the purchased products of a customer.





StockManager
ID
Tradingsstem: inventory:data relational algebra
query
D
Database
List<IOrderEntries>
Tradingsstem: inventory:app alternation of data
Report

Goal-Question-Metric plan



Applicability

Procedure applicable ?

Application on CoCOME

Modeling Language

Expressible by modeling language ?

Number of modeling elements for different types of dataprocessing

Number of missing modeling elements

Case Study

Access rights well defined ?

Number of covered access rights criteria:
- Aspect-oriented
- Positive
- Need-to-know
- Clear
- Concise
- Fundamental
- Efficient

Usable for different problem statements ?

Number of covered information flow classes:
- Illegal information flow
- direct flow between different roles
- information flow from high to low
- observable information flow

Evaluation for the quality of the access rights

- Evered and Bögeholz defined seven criteria to measure the quality of access rights

Access Rights	fulfilled ?
Specification	
Aspect-oriented	✓
Positive	✓
Need-to-know	✓
Comprehension	
Concise	?
Clear	?
Realization	
Fundamental	n/a
Efficient	n/a

Evaluation of covered information flow classes

- Problem statement: Non-influence = non-interference + non-leakage (Oheimb 2004).
  - Non-interference: High data inputs in the program flow has no effect on low data outputs.
  - Non-leakage: Unobservable that certain actions have taken place.

Data flow	fulfilled
Illegal information flow	✓
Information flow from high to low	✓
Direct information flow between roles	✗
No observable information flow	✗

Evaluation of covered information flow classes

- Problem statement: Non-influence = non-interference + non-leakage (Oheimb 2004).
  - Non-interference: High data inputs in the program flow has no effect on low data outputs.
  - Non-leakage: Unobservable that certain actions have taken place.

Data flow	fulfilled
Illegal information flow	✓
Information flow from high to low	✓
Direct information flow between roles	✗
No observable information flow	✗

Threats to validity

Internal Validity	External Validity	Construct Validity	Conclusion Validity
II, III	I	II	III

- I: Not applied to various systems.
- II: Not all criteria for good access rights are checked.
- III: Not all information flow classes are covered.

Threats to validity

Internal Validity	External Validity	Construct Validity	Conclusion Validity
II, III	I	II	III

- I: Not applied to various systems.
- II: Not all criteria for good access rights are checked.
- III: Not all information flow classes are covered.

Future work

- Method
  - Create a case study for the complete CoCoME system.
  - Apply the method to other systems (e.g Travelsystem (Katkalov et al. 2013)) and create further case studies.
- Case study
  - **short term work**
    - Evaluate the criteria *concise* and *clear*.
    - Define additional scenarios to cover all information flow classes.
  - **long term work**
    - Evaluate the criteria *fundamental* and *efficient*.
    - Definition of further information flow classes other than non-influence out.
    - Using the case study for a data based privacy analysis.

PIBA

- Problem
  - Usable case studies for data-based privacy analysis (DBPA) are difficult to create.
- Idea
  - Introduce a method for creating usable case studies for DBPA approaches.
- Benefit
  - Comparability for different privacy analysis approaches.
- Actions
  - Create a method for the creation of case studies.
  - Apply the method to a system.
  - Evaluate the created case study.

References I

Jakub Breier. “Asset Valuation Method for Dependent Entities”. In: *J. Internet Serv. Inf. Secur.* 4.3 (2014), pp. 72–81. URL: http://isyou.info/jisis/vol4/no3/jisis-2014-vol4-no3-05.pdf.

Mark Evered and Serge Bögeholz. “A Case Study in Access Control Requirements for a Health Information System”. In: *ACSW Frontiers 2004, 2004 ACSW Workshops - the Australasian Information Security Workshop (AISW2004), the Australasian Workshop on Data Mining and Web Intelligence (DMWI2004), and the Australasian Workshop on Software Internationalisation (AWSI2004) . Dunedin, New Zealand, January 2004*. 2004, pp. 53–61. URL: http://crpit.com/confpapers/CRPITV32Evered.pdf.

References II

- Jan Jürjens. "UMLsec: Extending UML for Secure Systems Development". In: *UML 2002 - The Unified Modeling Language, 5th International Conference, Dresden, Germany, September 30 - October 4, 2002, Proceedings*. 2002, pp. 412–425. DOI: 10.1007/3-540-45800-X_32. URL: https://doi.org/10.1007/3-540-45800-X_32.

- Kuzman Katkalov et al. "Model-Driven Development of Information Flow-Secure Systems with IFlow". In: *International Conference on Social Computing, SocialCom 2013, SocialCom/PASSAT/BigData/EconCom/BioMedCom 2013, Washington, DC, USA, 8-14 September, 2013*. 2013, pp. 51–56. DOI: 10.1109/SocialCom.2013.14. URL: https://doi.org/10.1109/SocialCom.2013.14.

References III

David von Oheimb. “Information Flow Control Revisited: Noninfluence = Noninterference + Nonleakage”. In: *Computer Security - ESORICS 2004, 9th European Symposium on Research Computer Security, Sophia Antipolis, France, September 13-15, 2004, Proceedings*. 2004, pp. 225–243. DOI: 10.1007/978-3-540-30108-0_14. URL: https://doi.org/10.1007/978-3-540-30108-0%5C_14.

Per Runeson and Martin Höst. “Guidelines for conducting and reporting case study research in software engineering”. In: *Empirical Software Engineering* 14.2 (2009), pp. 131–164. DOI: 10.1007/s10664-008-9102-8. URL: https://doi.org/10.1007/s10664-008-9102-8.

References IV

Stephan Seifermann. “Architectural Data Flow Analysis”. In: *13th Working IEEE/IFIP Conference on Software Architecture, WICSA 2016, Venice, Italy, April 5-8, 2016*. 2016, pp. 270–271. DOI: 10.1109/WICSA.2016.49. URL: https://doi.org/10.1109/WICSA.2016.49.

Evaluation Modeling language

Meta model	possible ?
relational algebra	yes
I/O operations	yes
Transmission of data	yes
Change of access rights	yes
Alternation of data	yes
ACM in system model	no

Operations matrix complete

Types of data processing	customer	account	P& S	system
Webfrontend	transmit	transmit	I/O transmit	n/a
PickupShop	transmit	transmit	I/O, transmit	n/a
Tradingsystem: inventory:app	change transmit	change transmit	change	n/a
Tradingsystem: inventory:data	rel. algebra operations	rel. algebra operations	rel. algebra operations	change
Tradingssystem: cashdeskline	change transmit	non-existent	change transmit	n/a

Definition of the value of an asset

- Different assets in system are related to each other.
- The assets are categorized in different levels. The value of an asset to the system is decreasing with descending numbers.
- A higher level is more crucial to protect for the system than the lower levels.
- In CoCoME:
  - Level 1: Customer and account data
  - Level 2: System and P& S data

Conclusion of the Procedure

- In the current state, we would argue it depends on the use of the resulting case study.
- Conclusion of the procedure
  - Access rights:
    - Concluded, further fulfillment of the criteria were not possible due to time constraints.
  - Information flow classes
    - If the covered information flow classes are sufficient for the intended use use of the case study
- No Conclusion of the procedure
  - Information flow classes are not covered yet.