2013 IEEE 37th Annual Computer Software and Applications Conference

UML based Security Function Policy Verification Method for Requirements Specification

Atsushi Noro
Graduate School of Engineering Division of
Electrical Engineering and Computer Science,
Shibaura Institute of Technology
307 Fukasaku, Minuma-ku, Saitama-City, Saitama 337-8570, Japan
ma12082@shibaura-it.ac.jp

Saeko Matsuura
Graduate School of Engineering Division of
Electrical Engineering and Computer Science,
Shibaura Institute of Technology
307 Fukasaku, Minuma-ku, Saitama-City, Saitama 337-8570, Japan
matsuura@se.shibaura-it.ac.jp

***Abstract*— One key to success for high quality systems developments is to verify not only functional requirements but also the security requirements at the early stage of developments. However, it is difficult for general developers who have only less security knowledge to define verifiable requirements specification without leakages and errors. To reduce these some leakages or errors, this paper proposes a UML-based security requirements verification method using the security knowledge of Common Criteria.**

Keywords—Commmon Criteria; Model Checking; Security Requirements; Verification; UML;

I. Introduction

In requirements analysis of a system required by various users, there are such requirements as access flow control that causes complicated analysis. Therefore, some leakages or errors might be occurred in these analyses because these are complex works and needed the developers' security knowledge. We have proposed a method of model-driven requirements analysis using UML [1]. The product of this method is called a requirements analysis model (RA model). This paper proposes a security requirements verification method for RA model using a Security Function Policy (SFP) as shown in Figure 1.

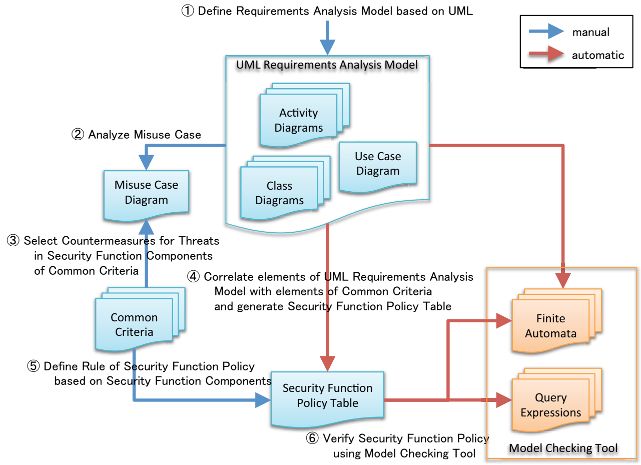
① Define Requirements Analysis Model based on UML
manual
automatic
UML Requirements Analysis Model
Activity Diagrams
Use Case Diagram
Class Diagrams
② Analyze Misuse Case
Misuse Case Diagram
③ Select Countermeasures for Threats in Security Function Components of Common Criteria
④ Correlate elements of UML Requirements Analysis Model with elements of Common Criteria and generate Security Function Policy Table
Common Criteria
⑤ Define Rule of Security Function Policy based on Security Function Components
Security Function Policy Table
⑥ Verify Security Function Policy using Model Checking Tool
Finite Automata
Query Expressions
Model Checking Tool

Figure 1. Outline of our method.

II. Method

A. Define UML Requirements Analysis Model

Both business flows and the entity data which are required to execute the target use cases are defined by activity diagrams and a class diagram in UML. An activity diagram specifies not only normal and exceptional action flows but also data flows which are related with these actions. An action is defined by an action node and data is defined by an object node being classified by a class which is defined in a class diagram. Accordingly, these two kinds of diagrams enable us to specify business flow in connection with the data.

B. Analyze the Misuse Case

To analyze threats against the target system, misuse case analysis [2] is introduced. A Misuse Case can express some use cases that should not happen by the malicious users. First, we reveal what are the assets to be protected from the malicious users based on the UML requirements analysis model. Secondly, against some use cases which deal with the specified assets, we extract some misuse cases acted by the malicious users. Here, an asset corresponds to an entity data defined by a class.

C. Select Countermeasures for Threates

Next, we decide each countermeasure against the misuse case by selecting an appropriate component in Common Criteria which is an international standard for computer security certification (CC [3]). It is structured by classes, families, and components. Classes are divided into such 10 familiar categories as User Data Protection, Privacy, and etc. Therefore, it is not difficult for a developer to select an appropriate class, so that he/she can define the concrete countermeasure according to the selected class.

D. Generate Security Function Policy Table

To implement the selected security functional component in CC, we define a SFP which represents the rules that the target system must enforce. SFP must specify its scope of control, by defining the subjects, objects, resources or information, and operations to which it applies. A rule in SFP

0730-3157/13 $26.00 © 2013 IEEE
DOI 10.1109/COMPSAC.2013.141

TABLE I. SECURITY FUNCTION POLICY TABLE

Subject		Object		Operation		Rule					
Name	Security Attribute	Name	Security Attribute	Activity Diagram	Action	FDP_ACF.1	FDP_IFF.1	FMT_MSA.3	FMT_MSA.1	FPR_ANO.1	･･･
Student	Role(Student ID)	⋮		⋮	⋮						
		Topic	Public/Private	select topics(Student)	Show <selected TopicHistry>		rule B				
				read topics(Student)	Show <selection topic>						
				contribute questions	Create Topic			rule C			
		Contributor	Role	select topics(Student)	Get current user						
				read topics(Student)	Show <selection topic>					rule G	
				contribute questions	Get Contributer						
		Attachment	Public/Private	read topics(Student)	Download Attachment	rule A					
				contribute questions	Register Attachment			rule C			
Teacher	Role(Teacher)	⋮		⋮	⋮						
		Topic	Public/Private	select topics(Teacher)	Show <selected TopicHistry>						
				read topics(Teacher)	Show <selection topic>						
				answer questions	Get selected Topic used Question number						
				answer questions	Add Answer and Upload Topic						
					Change Public/Private to Public				rule E		
					Change Public/Plivate to Private				rule F		
		Contributor	Role	select topics(Teacher)	Get current user						
				read topics(Teacher)	Show <selection topic>						
				answer questions	Get Contributer						
		Attachment	Public/Private	read topics(Teacher)	Download Attachment						
				answer questions	Register Attachment			rule D			
					Change Public/Private to Public				rule E		
					Change Public/Plivate to Private				rule F		

is defined by the relation between a subject, an operation and the target object. A subject carries out an operation. UML RA model consists of actors, use cases (activity diagrams), classes, and actions in the activity diagram. So we define the following mapping rule: a subject corresponds to an actor, an object corresponds to a class, and an operation corresponds to an action in a use case. According to this correspondence, Table I can be automatically generated from RA model except the security attributes and the rules.

E. Define Rule of Security Function Policy

Because SFP controls action flows by a rule based on security attributes, some new security attributes need to be defined against both the assets extracted in Section II-B and the subjects who carry out the controlled operation. On the automatically generated table, we define the rules of SFP based on subjects, objects, operations and selected security function components. Table I shows the example for defined Security Function Policy Table. Table II shows the detailed rules which written in Table I. The blanks in Table I means that any rule is unnecessary in the target system.

TABLE II. THE DETEAILED RULES

Rule A	executable Public/Private=Public or Contributer.Role=Subject.Role
Rule B	showable only Topic which Public/Private=Public or Contributer.Role=Subject.Role
Rule C	Create in Public/Private=Private
Rule D	Create in Public/Private=Private or Public/Private=Public
Rule E	Change Public/Private to Public
Rule F	Change Public/Private to Private
Rule G	Must not show if Subject.Role!=Contributer.Role

F. Verification using Model Checking Tool

Model checking is a technique to automatically verify a model by checking all paths exhaustively [4]. We use a model checking tool named UPPAAL [5] to verify that the requirements analysis model meets the specified SFP.

It is generally difficult to define appropriate model and query expressions so that developers can easily use a model checking tool. However, in our method, the target model which will be checked can be automatically transformed from the UML RA model. Moreover, query expressions can be derived from the specified rules in SFP table. For example, by Rule C, we can get a query expression for the following: Whenever the use case "contribute questions" reaches the end, the security attribute of "Topic" object should be "Private".

III. CONCLUTION

UML is a popular tool for specifying requirements, but it has less formality than formal specification techniques. We could define rules on appropriate security requirements by using the above-mentioned such specified elements in requirements analysis model in UML by corresponding to elements in CC so as to make it possible that requirements specification is verifiable.

REFERENCES

[1] S. Ogata. and S. Matsuura, “A Method of Automatic Integration Test Case Generation from UML-based Scenario,” WSEAS TRANSACTIONS on INFORMATION SCIENCE and APPLICATIONS, Issue 4, Vol.7, pp.598-607,2010.

[2] Sindre, G and Opdahl, A. L. “Eliciting security requirements with misuse cases”, Requirements Engineering Journal, Vol.10, No.2 (2005).

[3] Common Criteria, ”CC/CEM v3.1 Release4”: ISO/IEC 15408, http://www.commoncriteriaportal.org/cc/

[4] Y. Aoki, S. Ogata, H. Okuda and S. Matsuura, Quality Improvement of Requirements Specification Using Model Checking Technique, Proc of ICEIS 2012, Vol.2,pp401-406, 2012.

[5] UPPAAL, http://www.uppaal.com/, 2010.