How security requirements should be reused

Man Fai Ma

ID: 2538007

Computer Science Department

University of Auckland

mma004@ec.auckland.ac.nz

Abstract

Security requirement reuse is a powerful approach to help new security engineer without enough training to model a successful and reasonable security engine for system. This term paper would talk about the basic idea of system security engineering at beginning. Next explain what requirements reuse is. Then, the paper will refer to two approaches to show how requirements reuse applies to those approaches. And finally point out about the advantages and disadvantages of those two approaches.

1. Introduction

Requirement reuse is a new approach for develop security system. [Firesmith 2004] explain the basic idea of requirement reuse. It is very helpful for new engineer to develop a new security system. But [Firesmith 2004] has not a whole approach build on requirement reuse. So, approaches in [Castano et al. 1994] and [Toval et al. 2002] have been discussed to show how requirement reuse apply in different approaches. The main idea of this term paper is show the ability of nowadays approaches of requirement reuse. In section 2, this paper will explain the basic idea of security engineering, requirements, the method of how to identify requirements and the basic idea of requirements reuse. In section 3 and 4, two approaches generate base on the idea of requirements reuse, new approach in [Castano et al. 1994] and SIREN in [Toval et al. 2002] will be explain. In

section 5, some advantages and disadvantages of those two approaches would be discussed.

2. Basic idea of Security engineering

2.1 Security engineering and requirements

Briefly, Security engineering means a development process to model an engine in system that protect valuable system asset under security threats, and a chance of any kinds of attacks may happen from any kinds of attackers. There are many of security engineering approach to model and develop a new security system. Security requirement reuse is one of the successful approaches too. Security requirements are one of the main issues of any security engineering approach. Security requirements are set of functional requirements that they points out what security issue would be needed by systems. Those requirements always specify different levels of identification, authentication, authorization, integrity, privacy, etc for system should include under security need.

2.2 Identify and analysis security engineering

In [Firesmith 2004], author points out a very good model about how to identify what security requirements are requested. He used Assets-based risk-driven model in his research. Assets-based risk-driven model base on two main idea, identify assets need to protect and estimate risk if security fail. Briefly, the approach can separate to twelve parts, there are:

01 Identify Valuable Assets

02 Identify Likely Attacker Types

03 Identify Threats to these Assets

04 Determine Negative Impacts

05 Estimate and Prioritize Security Risks

06 Select Security Sub factor

07 Select Relevant Templates

08 Identify Relevant Functional Requirements

09 Determine Security Criterion

10 Determine Quality Measure

11 Determine Required Level

12 Specify Requirement

2.3 Security requirement reuse

[1][Firesmith 2004] stated a good statement to explain why requirement reuse: "The high potential reusability of security requirements is very beneficial because most requirements engineers have had no training in identifying, analyzing, specifying and managing security requirements and most requirements teams do no include subject matter experts in security." The reason of develop requirement reuse approaches is not for expert or professional. It mostly designs for new security engineer to improve quality and productivity of development. There a condition must be meeting before using reuse approach. Engineer must have some information of the previous model which existed and have same or similar domain interest with the current project. Because "reuse" method means must follow or build on another's model to develop. The reason for that work must exist because engineer can base on the information to prove the performance of that previous model. If engineer is an expert, he/she can create their own model rather than reuse another's model. Requirements reuse are also support some functions like analyzed, prioritized, and traced. Valuable and feasible are also proved when a good approaches likes section 3 and 4 applied. The reuse percentage can be very high showed in case study in [Toval et al. 2002]. There three categories of reuse approaches classified in [Cybulski & Reed 2000], there are text processing, knowledge management and process improvement. Text processing aims on the text of the requirement, base on understand the idea and focus the structure of the text format to process requirement reuse. Knowledge management aims on the problem modeling of requirement to process requirement reuse. Process improvement aims on change or improve the process in normal development approach to do include reuse and increase the productivity. All of three categories are useful on requirement reuse. Base on my understanding, the new approach in [Castano et al. 1994] belongs to knowledge management and SIREN in [Toval et al. 2002] belongs to process improvement. If all advantages in three categories combined may form a very successful

Table 2. Results of the application of SIREN for security requirements

Requirements layer	Chosen	Totals	% Chosen
Environment	111	136	81.6
Information system	91	121	75.2
Information	48	56	85.7
Organisation's functions	9	29	31
Intangible asset	7	10	70
Total	266	352	75.6

[1] Picture from [Toval et al. 2002].

approach for requirements reuse. [Cybulski & Reed 2000] also points out one possibility of combine all three categories. And prove requirements reuse can improve both productivity and quality of the resulting product.

3. New Approach for requirements reuse with schema

A new approach about how to reuse security requirement to model a security system has been discussed in [Castano et al. 1994]. In this approach (The paper did not give name for this approach, but I will call it “NA” below), NA’s theory is select numbers of candidate authorization schemas from existed system, then find out the difference between those schemas and the current system objects and roles to form a similarity set schema, and finally use the similar set schema to design the reusable security specifications. This approach also used one schema model called authorization schema.

3.1 Authorization model

Authorization schema has two types of authorization, user authorization and role authorization. User authorization is a pair form of user and role <u, r>. The pair means the user allowed to play the role in the same pair. For instance, <A, manager> means A can classify to be manager in the system. Role authorization includes role, object and action, <r, o, a>. It means r have right to execute a on o. For example, <manager, report, read> means manager can read report. There are one more form of role authorization, <r1, (o, a), r2> means r1 can give authorization to r2 for do a on o. For another example, <director, (letter, read), secretary> means director can give authorization to secretary for read letter.

3.2 Selection of candidate authorization schemas

To select candidate authorization schemas in the same domain of interest, this approach have two concern directions, relevance and quality. For relevance of the application, authors stated they choose a set of candidate schemas by its security characteristic. The domain of the schema must be same or very similar with the new application. For instance, schema of government application or personal information database domain would fit passport application. For Quality and completeness of the authorization schemas, authors stated they choose the schemas model significant number of roles and operations should be chose. They limited the schemas really significant and reduce the reuse cost would be chose.

3.3 Selection and classification of security specifications

When select and classify security specifications, engineers identify and grouping the elements in schema for next step to starting define reusable specifications. The idea is find the determine commonalities of elements within the chosen schema. Similarity between objects and similarity between roles would be defined in the step. To define between objects, authors fine out the objects have similar names with similar attributes. For example, object o1 and o2 have the same name “report” and both have similar attribute like edition date, topic and author. Similarity between roles is another issue need to define. Roles have similar name, attribute and authorizations counted in this section. For example, Project manager and Division manager have similar name, attribute are mostly same and similar authorizations like read and approve.

3.4 Design of reusable security specifications

After selection of candidate authorization schemas and selection and classification of security specifications, engineer can form a similarity set schema by grouped similar level, role and object. And base on similarity set schema, engineer start to design reusable security specification. And fill the reusable security requirement template.

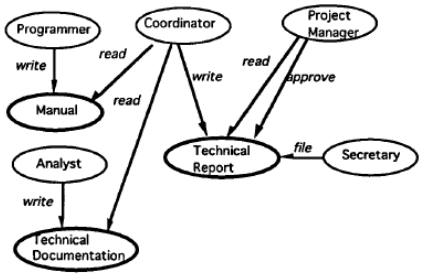

Figure 1: An example of authorization schema [2]

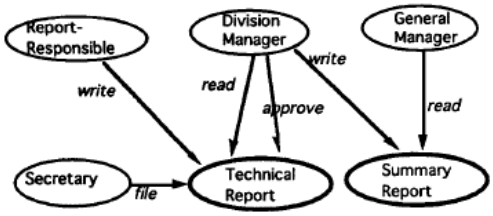

Figure 2: An example of authorization schema

Figure 3: An example of generic authorization schema [3]

[2] Picture from [Castano et al. 1994].
[3] Picture from [Castano et al. 1994].

4. SIREN

SIREN in [Toval et al. 2002] is one of the requirement engineer approaches which developed based on reuse security requirements. The target's of this approach is development process can be speeded up by minimizing modification.

4.1 Contents of the Requirements Repository

SIREN classify requirement in two types, there are parameterized and non-parameterized. Generally, parameterized requirement have some words in square bracket, for replaced by other words which meet the definition in square bracket.

Authors of [Toval et al. 2002] think, because traceability relationship between high-level requirements, requirement reuse can produce more benefit rather than pure design or code reuse only. SIREN has two types of traceability, inclusive trace and exclusive trace. When requirement A and B have inclusive trace relationship once requirement A has been reused, engineers would also reuse requirement B. On the other hand, if requirement A and B have exclusive trace relationship then if engineer can reuse requirement A or requirement B only.

4.2 Requirements Documents Hierarchy

By increase requirement reusability for SIREN, authors follow all most popular requirement engineer standard, and used those documents templates. They are System Requirements Specification (SyRS) (IEEE Std. 1233; IEEE Std. 12207.1), Software Requirements Specification (SRS) (IEEE Std. 830), System Testing Specification (SyTS), Software Testing Specification (STS) and Interface Requirements Specification (IRS) (IEEE Std. 830). With the standard specification above, requirements in SIREN would easier to reuse by other engineers. On top of this, SIREN can include instance, tables and schemas to improve its ability too.

4.3 Process Model

SIREN is a spiral model processing approach which started with two main steps: Requirements selection and Specific requirements elicitation. Requirements selection talks about reuse the old requirements, Engineers finds security need by asset-based and

risk-driven method, make selection of security requirements by its priority, obligement[4] level and development budget. Then engineer fill them in templates, show those templates to stakeholders and discuss with them about what requirements in template they want to include. Then, all the inclusive related requirements should be added, and all the exclusive related requirements should be removed. Specific requirements elicitation talks about create any new requirements for specific system. In the same meeting with stakeholder, engineers should build some informal requirements of problem domain by specific knowledge of the system. Once finish the above two steps, the reused requirement templates from requirements selection and informal requirements documents from specific requirements elicitation are written. The remaining work is discovering problems and negotiates stakeholders to remove redundancy and inconsistency, identify Agreed Requirements and then write Draft Requirements Documents. By repeat those steps above with stakeholders in spiral model, engineer can build up a Validated Requirements Documents and programmers can start the implantation.

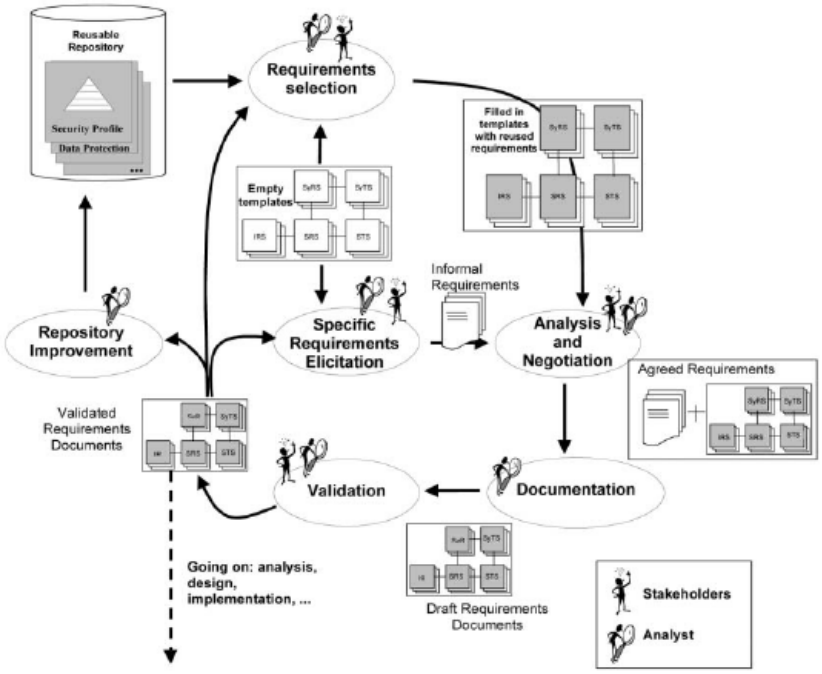

Fig. 9. SIREN approach for requirements reuse.

5

[4] Obligement (ô`blīj´ment) noun
Definitions: *Scotland* **favor:** a favor done for or owed to somebody else
[5] Picture from [Toval et al. 2002].

5. Advantages and disadvantages of requirements reuse approaches

5.1 NA (New Approach in section 3)

"NA" in section 3 is friendly to new engineer, because this model does not need any complex skill or special to follow. The authorization model is easy to handle, it has only 3 types of form and simple. The two selection steps also effective to less the unnecessary time on modeling and cost waste by good selection done. Select candidate authorization schemas less time waste on doing research in second selection. Selection and classification of security specifications can less the time waste on model no use objects and roles in implantation.

The biggest disadvantage of this approach is its process only up to security templates filled. And lack of case study or other information about what will do after reusable security templates filled. In the conclusion of [Castano et al. 1994], authors' point out their concern of the approach applicability with other approaches. They want this approach can cooperate with other approaches to improve its ability. Another disadvantage structure of authorization model similar with use case or OO model, but its less support include or extend in use case model and generalization in OO model. By the reason of authorization model using UML tools likes' structure, improve it to more close UML model can improve both of the users friendly and function for it.

5.2 SIREN

SIREN is a powerful approach of requirement reuse. In case study section in [Toval et al. 2002], it did assets and risk analysis to define the valuable assets and the risk if fail. Using priority and obligement[6] value to find out which requirements are more important. Use spiral model rather than waterfall model to increase the chance to community with stakeholder.

---

[6] Obligement (ô`blīj´ment) noun
Definitions: *Scotland* **favor:** a favor done for or owed to somebody else

Traceability is the one of the main advantage I can find for this approach. Traceability relationship of requirement in [Toval et al. 2002] means when engineers add a trace attribute about which requirements related with that requirements. For instance, if engineers add trace attribute for requirement called R1. With this attribute, engineers can less time spending on search out what requirements engineer would needed to model with R1, or any another requirements engineer can model once they modeled R1. Furthermore, programmers always base on requirement to program a system base on the requirements of that system. So, traceability in requirement reuse can speed up design and implementation too.

Base on the idea in [Toval et al. 2002], I can't find a main disadvantage of the approach, but I would said there some problem when read through this paper. Because the paper tells people many powerful function SIREN have and a case study section are also included, but I still feel confusing on some parts of paper. For example, the paper did not clear the idea in the process model section about how to choose requirements at the beginning. Although it add back some details in case study section, but it still not clear how to make discussion to give what level of priority and Obligement[7] value to requirements. And I feel base on the explanation of the paper, SIREN are still too hard for new security engineer.

6. Conclusion

In this term paper, I tried to point out the basic idea of security requirements reuse and some approaches build on requirements reuse. Requirement reuse is not a new environment, but I feel it is not a popular topic when I doing research. I think the reason is people not very interesting on a topic have no actual output and not very profitable. But base on the same reason, I think this environment still have lots of possibility for doing research and development. My personal feeling of requirements reuse is amazing. The idea of requirements reuse to help new engineer to model a more powerful security system is great. It is really helpful for student like us to model security system. I hope more research and development of software and tools for requirements reuse will happen.

---

[7] Obligement (ô`blīj´ment) noun
Definitions: *Scotland* **favor:** a favor done for or owed to somebody else

7. Reference:

[Firesmith 2004]
Firesmith, Donald G. “Specifying Reusable Security Requirements.”
Journal of Object Technology (JOT) 3, 1
(January/February 2004): 61-75.
<http://www.jot.fm/issues/issue_2004_01/column6>.

[Castano et al. 1994]
Silvana Castano, Giancarlo Martella, Pierangela Samarati
“A new approach to security system development”
Little Compton, Rhode Island, United States
Pages: 82 - 88
Year of Publication: 1994
ISBN:0-8186-6335-9
<http://portal.acm.org.ezproxy.auckland.ac.nz/ft_gateway.cfm?id=283865&type=pdf&coll=ACM&dl=ACM&CFID=58362163&CFTOKEN=67112292>

[Toval et al. 2002]
Toval, A., Nicolas,J., Moros,B., Garcia,F. "Requirements Reuse for Improving Information Systems Security: A practitioner Approach" Requirements Eng.. Journal Vol 6 pp206-219, 2002 Springer-Verlag

[Cybulski & Reed 2000]
Cybulsky J, Reed K. Requirements classification and reuse:
crossing domains boundaries. In: 6th international conference on
software reuse (ICSR’2000). Lecture Notes in Computer Science.
Springer, Vienna, 2000, pp 190–210