# Security in the Context of Dependability

Tacksoo Im
John D. McGregor
Clemson University
{tim,johnmc}@cs.clemson.edu

## Abstract

Security, as an architectural quality, is often thought to be measured in terms of availability, confidentiality and integrity. These qualities are part of a broader quality - dependability. There are inherent tradeoffs among the qualities that define security and dependability. Architectural tactics, or architectural design decisions, that enhance one aspect of dependability can decrease security and vice-versa. In addition, this is a multi-scale problem in that different quality attributes are measured on different scales of reference including some that are not quantitative. In this paper we present a qualitative approach to managing interactions among the attributes used to define security.

## Introduction

Defining a system architecture to support products that are secure requires the architect to address both functional and non-functional requirements. Quality-driven techniques require the architect to explicitly consider these non-functional requirements – referred to as quality attributes - during even the earliest architecture decisions. Techniques such as quality attribute workshops (QAW) and the architecture tradeoff analysis method (ATAM) [Kazman 00] are used to identify the desired quality attributes of a system but it is hard for the architect to design the system when several of the quality attributes interact with each other. That is, an architecture decomposition that enhances one attribute may degrade another. Managing the tradeoffs among qualities is hard for several reasons. It is hard for an architect to be knowledgeable about all the quality attributes that are involved in a particular system and some of the qualities, such as security, are not measured on quantitative scales while others are quantitative. There are reasoning frameworks that assist the architect in quantitatively analyzing quality attributes such as performance. But there are few techniques for reasoning about attributes that are qualitatively represented, such as security. In this paper, we present a qualitative approach to reasoning about security at the architectural stage.

## Background

Dependability is defined as the degree to which trust can be justifiably placed on a computer system. This is usually taken to include the qualities of reliability, availability, safety, integrity, confidentiality and maintainability. [Avizienis 00] This subsumes the definition of security usually taken to include: availability, integrity, and confidentiality. In our work on designing dependable systems we have identified four interactions among the qualities within dependability that involve qualities related to security.

- Availability vs. Confidentiality
  Design decisions that increase the availability of the system can decrease confidentiality by introducing prolonged exposure. [Warns 05]
- Availability vs. Integrity
  Design decisions that increase the availability of the system can decrease integrity by exposing data for longer periods of time to the possibility of malicious changes. [Warns 05]
- Safety vs. Confidentiality

Design decisions that increase safety can come in direct conflict with confidentiality because safety decisions often require distribution of knowledge. (e.g. replication prevents loss of data at the cost of a higher probability of theft and unauthorized modification) [Schneier 03]

- Safety vs. Integrity
  Design decisions that increase safety can come in direct conflict with integrity because safety decisions often require distribution of knowledge. [Schneier 03]

The architect must evaluate the tradeoffs between qualities when designing a dependable system. Analytically reasoning about composite qualities, i.e. qualities defined in terms of other qualities, that are measured on different scales, some of which are not quantitative, is not readily handled by existing techniques. Since security is not measured on a scale, a goal based scale will be used to support design reasoning. [McGregor 07] The goals on which these scales are based are called softgoals because there is no precise, objective definition of the goal or exact criteria for satisfying them. [Chung 00] A softgoal will not capture the level of detail found in performance models based on queuing theory but it will provide qualitative "indicators" that guide the architect. A softgoal is satisfactory for our purposes because qualitative reasoning techniques will allow us to make decisions about satisficing a softgoal but not optimizing it.

## Qualitative Reasoning about Security

Qualitative reasoning [Iwasaki 97] provides a means of making decisions involving attributes that can not be expressed quantitatively. Qualitative techniques do assume some type of ordinal scale. The reasoning rules use two fundamental characteristics: a current position on an ordinal scale and an indication of whether the attribute is changing its value and if so in which direction along the scale. For example, the security attribute of a piece of software might be rated on an ordinal scale as "very" secure and that recent architectural changes are making the software "more" secure. Qualitative reasoning supports building models that represent these relationships between qualitative values. These models support inferences about how the values change over time and how they cause other values to change.

For qualitative reasoning about security, the model must consider the direction of change for each quality and the inequality relationship that exists among tactics in relation to how each tactic influences the qualities that comprise security. This matrix of relationships is the minimum required to evaluate the impact of the choice of tactics on the system being designed. For example, as shown in Table 1, consider two tactics that improve security [Steel 05] and a tactic that improves availability. It is difficult to assess the net effect of these three tactics on the degree to which the resulting system is secure since relative magnitudes of the "-" effect of replication and the "+" effect of a validator can not be compared. ( ++ is strong positive satisficing, + is weak positive satisficing, -- is strong negative satisficing, - is weak negative satisficing) [Chung 95]

**Table 1 - Effect of Tactic on Quality Attribute [Warns 05, Steel 05]**

| | Availability | Confidentiality | Integrity |
|---|---|---|---|
| **Implementing a secure pipe** | No change | ++ | ++ |
| **Implement a intercepting validator** | ++ | ++ | ++ |
| **Replication of modules** | ++ | -- | -- |

The complexity in reasoning about these tactics is present partially because these attributes are not quantitative and partially because the measures are on different scales. It can be overwhelming to keep track of how each tactic influences each sub-quality of security and how each tactic relates to other

tactics. For this reason, we are developing a modeling technique to assist the architect in reasoning about security.

## Satisficing security requirements: An Example

The qualities that are of most importance to a financial web-service are the following:

1. Confidentiality
2. Integrity
3. Reliability
4. Availability

How will the use of the two tactics, implementing secure pipes and introducing replication affect the overall quality goals of the system? As shown in figure 1, the two tactics influence confidentiality and integrity in different directions, but the confidentiality and integrity of the overall system would have decreased after the application of the two tactics. This is because there is an inequality relationship between the tactics as it was determined that "percentage of replicated elements" has a greater impact on confidentiality and integrity than "percentage of secured pipes." (The application of these tactics results in an increase in the "percentage" of the replicated elements and secured pipes)

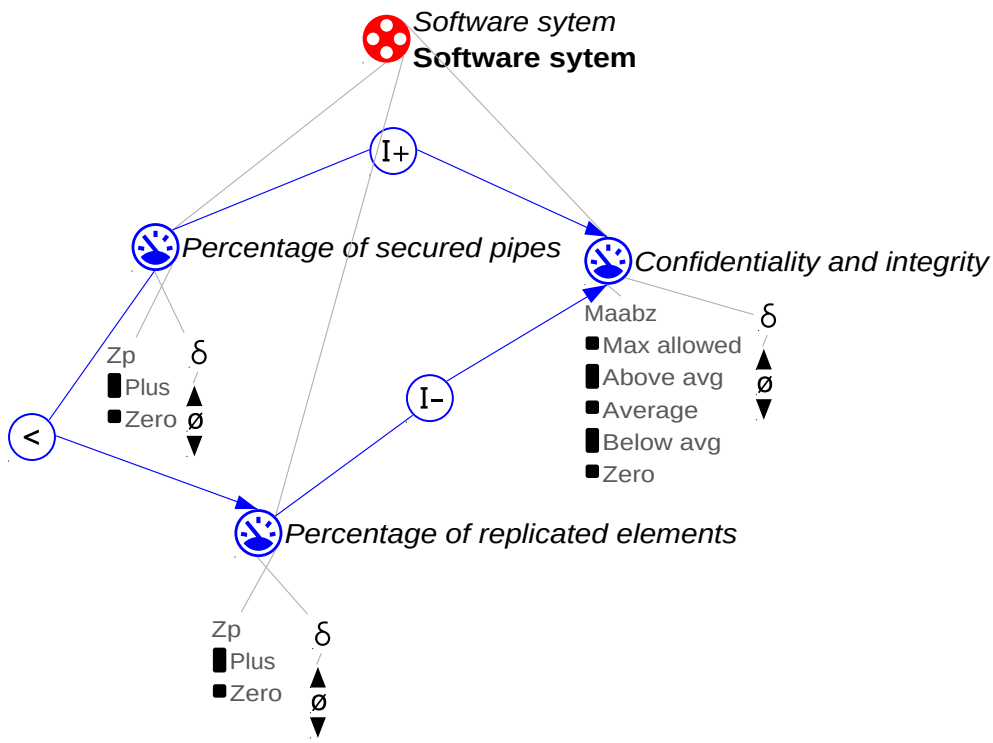


Figure 1

The inequality relationships between tactics are subject to change depending on the context of how they are applied and the architect has to decide on the relationship. The Garp3 tool used for qualitatively reasoning generates all possible cases if no inequality is specified. The power of the qualitative method comes when we combine these model fragments (such as the two shown in figure 1) that contain these inequalities to reason how the overall security of the system changes in response to the tactics that have been applied to the system. In our example, the financial web-service firm has confidentiality and integrity as its top priority and therefore it is not advisable to implement replication as an architectural tactic for the system.

To facilitate the application of qualitative reasoning to security a causal/qualitative model of security is needed. This model will describe how each tactic influences the sub-qualities of security (availability, confidentiality and integrity) and provide a knowledge base for qualitatively reasoning about security. The knowledge base will also contain the necessary data for reasoning about security in a broader context such as dependability. Capturing the causal/qualitative model will be very much like capturing an ontology but with relationships such as relative orders of magnitude and inequalities.

## Conclusion

The qualitative method used to reason about security may, on the surface, seem too simplistic to be useful but research [Hastie 01] indicates that simple linear models are very accurate in supporting decision making and predictions. In this paper, we have presented how security architectural tactics can influence dependability and vice-versa. Certain security tactics can hinder the dependability goals of a system. We have also presented a reasoning method to choose architectural tactics that can help the architect achieve the quality goals of a system.

## References

[Avizienis 00] Algirdas Avizienis, Jean-Claude Laprie, and Brian Randell. Fundamental Concepts of Dependability.

[Chung 95] Chung, L.K. Nixon, B. and Yu, E. “Using Non-Functional Requirements to Systematically Select Among Alternatives in Architectural Design", Proc., 1st Int. Workshop on Architectures for Software Systems, Seattle, April 24-28, 1995., pp. 31-43.

[Chung 00] Chung, L.K. Nixon, B. and Yu, E. “Non-functional Requirements in Software Engineering", Kluwer Academic Publishers, 2000.

[Hastie 01] Hastie, R. and Dawes, RM. “Rational Choice in an Uncertain World: The Psychology of Judgment and Decision Making”, Sage Publications Inc , 2001 Thousand Oaks CA.

[Iwasaki 97] Iwasaki, Y. Real-world applications of qualitative reasoning, Expert, IEEE [see also IEEE Intelligent Systems and Their Applications], 1997, pp. 16—21.

[Kazman 00] Kazman, R. Klein, M. and Clements, P. ATAM: Method for Architecture Evaluation. CMU/SEI-2000-TR-004.

[McGregor 07] John D. McGregor and Tacksoo Im. A Qualitative Approach to Dependability Engineering, Proceedings of Dahstuhl Seminar #07031, January 2007.

[Schneier 03] Schneier, B. Beyond Fear: Thinking Sensibly About Security in an Uncertain World, Copernicus Books, 2003.

[Steel 05] Steel, C. Nagappan, R. and Lai, R. Core Security Patterns: Best Practices and Strategies for J2EE(TM), Web Services, and Identity Management, Prentice Hall, 2005.

[Warns 05] Warns, Timo Engineering Intrusion-Tolerant Software Systems. In: Dagstuhl Workshop, 22 - 25 May 2005, Dagstuhl, Germany.