Aggregation of security attributes based on the granularity level of the system

Thesis Submitted in Partial Fulfilment of the Requirements for the Degree

of

Master of Science

to the Department of Computer Science of Freie Universität Berlin

by

Artemij Voskobojnikov
Student ID: 4557770
voskobojnikov.artemij@gmail.com

First Reviewer: Prof. Dr. Jörn Eichler
First Reviewer: Prof. Dr. Marian Margraf

Berlin, <Datum>

Affirmation of independent work

I hereby declare that I wrote this thesis myself without sources other than those indicated herein. All parts taken from published and unpublished scripts are indicated as such.

Berlin, <date>

(Artemij Voskobojnikov)

Acronyms

List of Figures

Abstract

Contents

1 Introduction

Security concepts can be used to capture the interacting system components, potential threats and countermeasures.
For large information systems such concepts can become very large because of the number of the involved sub-systems/components. Interconnectivity and interdepence amongst components may increase the overall system complexity and it might be therefore difficult to detect all potential impacts [2]. Methods for system abstraction that address this problem already exist. The abstraction here is the creation of representation layers which only reflect relevant properties of a system and therefore provide a better level of understanding for the respective user [9]. This for example can be achieved by different projections which reflect different granularity levels of a system displaying different levels of details [11].
In the security context such projections could be used to focus on the security or insecurity of certain sub-systems. Security attributes of components could thus be viewed separately and the security risk for a respective component could be derived. This might become especially useful when the security concept is incomplete or only partially available. An aggregation of security attributes based on the chosen projection will become possible. The system structure could then be used to derive security attributes for components that previously had none. Thus, potentially new information might become processable.
Aggregation methods for security attributes have already been suggested by researchers, e.g. transformation rules for security requirements by Menzel et al. [7] or aggregation rules for attack graphs by Noel et al. [8]. None of those methods take granularity levels or general system hierarchy into account whereas the goal of this thesis is to provide an approach which makes it possible for a user to select a sub-system of interest, i.e. a projection which reflects a certain granularity level and provides the corresponding security attributes. The relevant attributes as well as dependencies and possible aggregations will be shown to ensure an overall complete picture of the selected sub-system. This information can then be used to assess and improve the security level of the selected projection or its dependencies.

2 Background

Prior to addressing the actual approach and implementation some concepts and terms have to be introduced. Firstly, the term *security concept*, as it is used throughout the thesis, is being described. A definition of *granularity levels* and system abstraction follows. Lastly, a section covers *model transformations* and *aggregation rules* on security attributes.

2.1 Security

Many different definitions of security exist. Here, a slightly adapted definition of a *information security management system* (ISMS), as it is found in the ISO/IEC 27001 [1], is being used.

> *„The information security management system preserves the confidentiality, integrity and availability of information by applying a risk management process and gives confidence to interested parties that risks are adequately managed“*

It is also added that the ISMS is integrated into the overall management structure and is vital for many of the organizational processes [1].
The definition above only covers information security, which in the scope of this thesis, is insufficient. Here, we define security as the preservation of confidentiality, integrity and availability of assets, where assets can be either physical or logical.

2.2 Enterprise Security

To define the term *security concept* one has to look at the architecture of enterprises to understand the interconnectivity and interdependence between services, security being one of them.

2.2.1 Enterprise Architecture

Information systems tend to be a very complex artifacts that combine different views and requirements from various stakeholders of different backgrounds [6].
Software, IT platforms and IT related goals in general are covered in an *Information System Architecture* (ISA). ISA does not take any business-driven influences into account and is therefore insufficient when describing the complex dependencies in corporations, especially when it comes to security as described in Subsection 2.2.2.

Enterprise Architecture Modeling tries to overcome such possible difficulties and combines IT related concerns with business and organizational goals and shows possible interrelationships. It therefore provides an approach for an improved understanding of complex enterprise processes [12]. The Federal Deposit Insurance Corporation (FDIC) published the results of an audit of its own implementation of E-Government principles [4] and their division of information technology in Figure 1 depicts the interrelations very well.

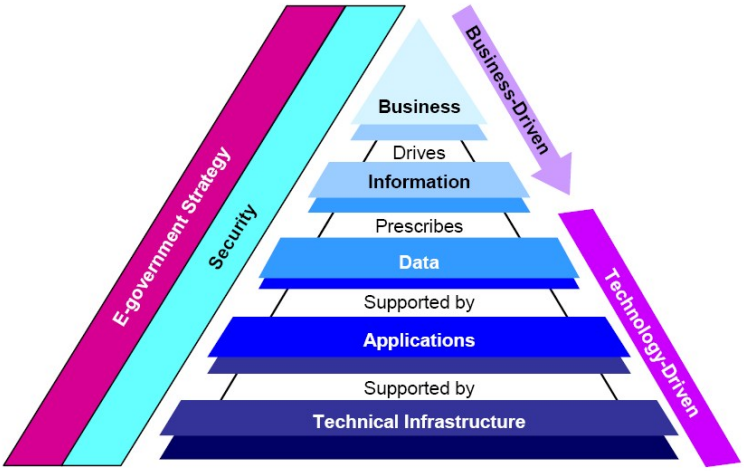

Figure 1: Division of Information Technology by FDIC

2.2.2 Security Architecture

Information Security has often been merely an afterthought in corporations [3] until a concept of a *Security Architecture*, published in a whitepaper by The Gartner Group [5], was introduced. According to [5] an *Enterprise Information Security Architecture* (EISA) is an essential tool for improving security processes in corporations. EISA principles stand in a direct relationship with the EA principles and should be validated against them [5]. To highlight this relationship security considerations during phases of the *The Open Group Architecture Framework Architecture Development Method* (TOGAF ADM), which is shown in Figure 2, will be briefly described.

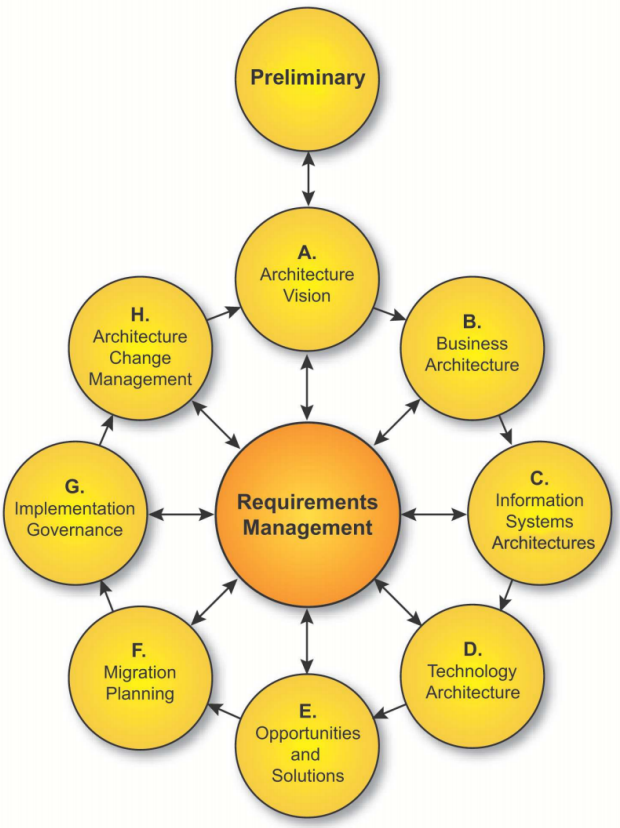

Figure 2: Overview of the phases of TOGAF

Security concerns can be found throughout the TOGAF phases which hints at the overall importance of security in corporations. TOGAF combines four architecture domains, the *Business Architecture*, the *Data Architecture*, the *Application Architecture* and the *Technology Architecture*. The following section will depict security considerations of two domains to show possible interrelations.

During the *Business Architecture* phase in the ADM actors and handlers of the system have to be identified. Costs and potential incoveniences because of security measures have to be assessed as well. In general one can say that the impacts of security/insecurity on the business/product are being highlighted. It is tried to put an emphasis on security as early as possible to prevent costly changes in later phases in the ADM.

During the *Information System Architecture* phase the classification levels of processed data have to be determined and documented. Direct dependencies to the *Business Architecture* are also listed, e.g. the identification of information lifespan according to business goals and regulations.

Similar relations can be found for security considerations from various phases of the ADM. This once again shows the overall presence of security and the high level of complexity within an enterprise.

2.2.3 Common Criteria

The overview of TOGAF showed the importance of security in corporations. The following section will present a way of modeling security concerns for an asset of interest.
Common Criteria proposes an evaluation by using a so called *Security Target* (ST), a construct that encapsulates the *Target of Evaluation* (TOE), threats to the TOE and countermeasures [10]. The goal of the evaluation is to show that the used countermeasures are sufficient to counter potential threats and thus implying that the TOE is sufficiently protected.

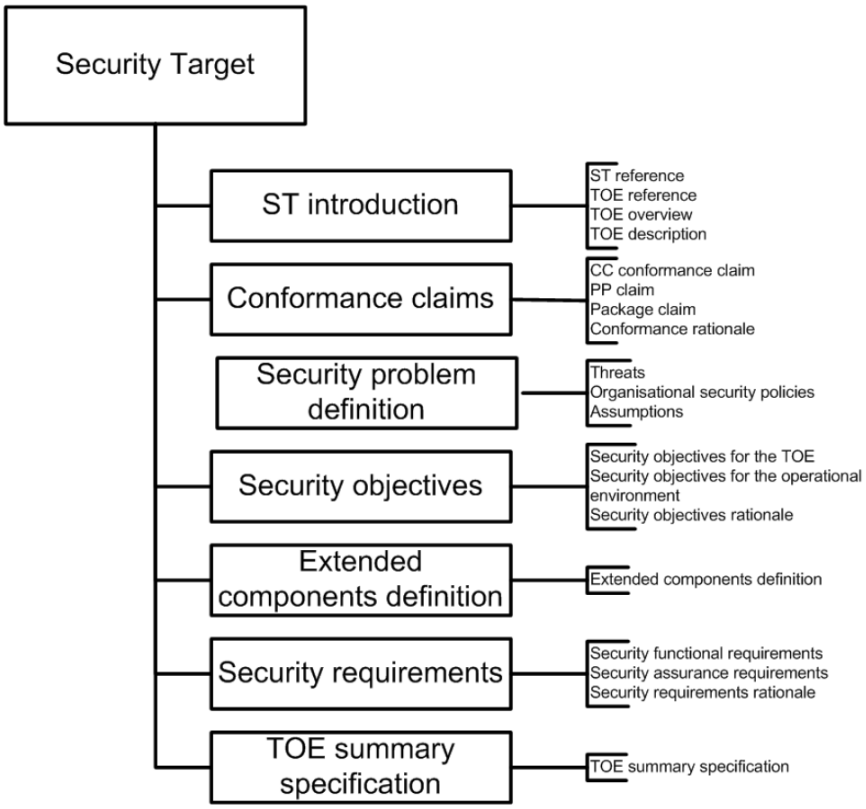

Figure 3: Overview of the Security Target contents

A description of all the contents of a ST is unnecessary here and only the key security attributes of a ST that will be used to construct a *Security Concept*

(Subsection 2.2.4) are being introduced.
The *Security Problem Definition* defines, as the name suggests, the security problem that is being addressed. Apart from containing guidelines and assumptions it contains *Threats* which are *„[...] adverse actions performed by a threat agent on an asset“* ([1], p. 66).
A *Security Objective* is an abstract solution to the previously defined security problem. There exists a possibility to divide the *Security Objectives* into part wise solutions, one being the *Security Objectives for the TOE* and the other being the *Security Objectives for the Operational Environment*. Moreover does the ST contain traces showing which objectives address which threats, guidelines and assumptions and a set showing that all threats, guidelines and assumptions are addressed by the security objective.
Security Functional Requirements (SFR) are a more detailed translation of the previously defined *Security Objective*. Despite being more detailed, SFR have to be still independent from specific technical solutions.
Lastly, STs contain a TOE summary specification where it is stated how the TOE meets all the SFRs and how exactly those requirements are met on a technical level.

2.2.4 Security Concept

The term *Security Concept*, as it is defined here, is based on the constructs introduced in the previous chapters, namely *Security Architecture* and *Security Target*. An overview follows.
Assets are the to be secured objects of interest, i.e. TOE according to Common Criteria. *Assets* can be either logical or physical and can be grouped to sets, if needed.
Security Goals (SG) is the equivalent to the *Security Objective*. A valid SG must address an *Asset* and a *Security Goal Class* that defines the actual purpose of the SG. In general the set of *Security Goal Classes* consists of *Confidentiality*, *Integrity* and *Availability* but can also be expanded by further classes such as *Authenticity*.
Threats serve the same purpose as proposed by Common Criteria. They are adverse actions performed by an entity against an *Asset*.
This information is all brought together in *Security Requirements* that are defined in natural language and show the interrelationships between elements. A *Security Goal* has to be mentioned as well as an *Asset* and a *Threat* against which the object of interest should be protected.
Lastly, *Controls* are the technical measures that counter or minimize the *Threats*.

The following table depicts the relationships between all the security attributes:

Name	Contains	Description
Asset	-	Digital or physical object of interest that should be secured
Security Goal Class	-	Defines the purpose of the Security Go
Security Goal	Security Goal Class, Asset	Defines the security objective
Threat	Asset	Adverse action against an Asset
Security Requirement	Asset, Security Goal, Threat	Security Objective in natural languag
Control	Threat	Measure to minimize or mitigate the Th

2.3 Granularity Levels

2.4 Model Transformation

2.4.1 Allocation

2.4.2 Aggregation rules

References

[1] ISO/IEC 27001:2005 - information technology – security techniques – information security management systems – requirements. Technical report, 2005.

[2] Mark Branagan, Robert Dawson, and Dennis Longley. Security risk analysis for complex systems. pages 1–12, 2006.

[3] Edwin Covert. Using enterprise security architectures to align business goals and it security within an organization. http://www.ansfederal.com/, 2010.

[4] FDIC. Implementation of e-government principles. https://www.fdicig.gov/reports05/05-018-508.shtml, 2005.

[5] Gregg Kreizman and Bruce Robertson. Incorporating security into the enterprise architecture process. *Gartner Research*, 2006.

[6] Stephan Kurpjuweit and Robert Winter. Viewpoint-based meta model engineering. In Manfred Reichert, Stefan Strecker, and Klaus Turowski, editors, *Enterprise Modelling and Information Systems Architectures - Concepts and Applications, Proceedings of the 2nd Int'l Workshop EMISA 2007*, volume P-119, pages 143–161, Bonn, Oktober 2007. Gesellschaft für Informatik, Köllen.

[7] Michael Menzel, Christian Wolter, and Christoph Meinel. *Towards the Aggregation of Security Requirements in Cross-Organisational Service Compositions*, pages 297–308. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[8] Steven Noel and Sushil Jajodia. Managing attack graph complexity through visual hierarchical aggregation. In *Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security*, VizSEC/DMSEC '04, pages 109–118, New York, NY, USA, 2004. ACM.

[9] Klaus Pohl, Harald Hönninger, Reinhold Achatz, and Manfred Broy. *Model-Based Engineering of Embedded Systems: The SPES 2020 Methodology*. Springer Publishing Company, Incorporated, 2012.

[10] The Common Criteria Recognition Agreement Members. Common criteria for information technology security evaluation. http://www.commoncriteriaportal.org/, September 2012.

[11] Judith Thyssen, Daniel Ratiu, Wolfgang Schwitzer, Alexander Harhurin, Martin Feilkas, and Eike Thaden. A system for seamless abstraction layers for model-based development of embedded software. In *Software Engineering (Workshops)*, pages 137–148, 2010.

[12] R. Winter and R. Fischer. Essential layers, artifacts, and dependencies of enterprise architecture. In *2006 10th IEEE International Enterprise Distributed Object Computing Conference Workshops (EDOCW'06)*, pages 30–30, Oct 2006.