Catalog of Security Tactics linked to Common Criteria Requirements

CHRISTOPHER PRESCHERN, Institute for Technical Informatics, Graz University of Technology

Security tactics describe security design decisions in a very general, abstract, and implementation-independent way and provide basic security design guidance. Tactics directly address system quality attributes and can be seen as building blocks for design patterns. In order to establish a more detailed security tactic collection, we link them with the Common Criteria security certification standard by establishing a connection between the security tactic goals and the Common Criteria Security Functional Requirements through Goal Structuring Notation. In this paper we give a brief introduction to the Common Criteria standard and to Goal Structuring Notation, we present the full structured and refined catalog of security tactics, and we discuss benefits of the link with the Common Criteria security standard regarding security certification.

Categories and Subject Descriptors: D.2.11 [**Software Engineering**]: Software Architecture—*Patterns*; K.6.5 [**Management of computing and information systems**] Security and Protection

ACM Reference Format:

Preschern, C. 2012.Catalog of Security Tactics linked to Common Criteria Requirements. jn 0, 0, Article 0 (0000), 16 pages.

1. INTRODUCTION

Security certification allows to evaluate a system regarding its overall security by providing a pool of requirements which have to be fulfilled. The security certification process, however, does not provide any methods to evaluate the influence of single design decisions during system development. The link between single architectural design decisions and their effect on the security quality attribute is described by [Bass et al. 2003] as security tactics. Architectural tactics, in general, address a single system quality attribute and are more general and implementation independent than design patterns, which can be composed of tactics [Kumar and Prabhakar 2010].

To evaluate the effect of architectural design decisions on security certification, a link between security tactics and the certification process is required. In this paper, we establish this link by mapping requirements given in the Common Criteria security standard to security tactics. We analyze Part 2 of the Common Critera standard which contains Security Functional Requirements (SFRs) and relate them to security tactics using Goal Structuring Notation. With this link between Common Criteria security certification and security tactics we refine security tactics. Additionally, we refine the tactics by structuring them and by gathering information from literature about their consequences and related tactics. We present the full catalog of security tactics and discuss the benefit of the established link to Common Criteria SFRs. Our security tactics catalog brings the advantage that the tactics are more structured and can provide a system architect with more detailed information about the effect of security tactics on the security quality attribute and on security certification.

Section 2 of this paper presents related work on security tactics, especially on their link to security certification. Section 3 gives an introduction to the Common Criteria security standard and Section 4 introduces Goal Structuring Notation. In Section 5 we present the full security tactics catalog with focus on the link of the security tactics to Common Criteria requirements. In Section 6 we discuss the soundness of the mapping and Section 7 concludes this work.

2. RELATED WORK

Architectural tactics are introduced by [Bass et al. 2003]. They cover tactics addressing the quality attributes availability, modifiability, performance, security, testability, and usability. This collection of tactics is extended by tactics addressing the quality attribute safety [Wu 2003] and by refinement of availability tactics [Scott and Kazman 2009]. Security tactics are extended in [Wyeth 2009], where a formal specification for them is defined. This allows to formally prove the implementation of security tactics. [Kim et al. 2009] discusses security tactics and their relationship to each other and to other tactics. They present the relationships between the tactics through feature modeling notation. An empirical study on the relationships between architectural tactics given in [Al-Daajeh et al. 2011] where the effect of safety tactics on quality attributes including security is covered.

Ryoo et al. suggest to extend security tactics by mining existing security patterns in order to find general tactics, but he do not actually extend the security tactic catalog. They also give requirements which have to hold for design decisions in order to be considered as a tactic. Tactic are domain neutral and are not attached to a particular problem, they cannot be divided into multiple tactics, and they just address a single quality attribute [Ryoo et al. 2010]. Another approach related to security patterns which uses mining is presented by Schumacher [Schumacher 2002] who suggests to use security certification standards in order to mine for security design patterns. Building patterns out of standards has the advantage, that the security standard is well accepted by a huge community and its requirements are more likely to be complete than security requirements developed by an individual. Therefore the standard allows to build patterns on a well matured basis. In [Schumacher 2003], the SFR of the Common Criteria standard are taken as input to discuss the forces affecting an architectural security pattern. Security patterns consisting of SFRs are also suggested in [Bialas 2011a] and [Bialas 2011b], where the security development process is addressed in particular. A semi-formal way to trace security tactics along the security development process is presented in [Houmb et al. 2009] where Common Criteria requirements are are modeled with UMLsec.

[Wu 2007a] analyzes Common Criteria SFRs of existing products in order to reason about the effect of these requirements on system quality attributes. He also constructs SFR requirement patterns for different domains (e.g SFR patterns suitable for operating systems). The aim of his work is to provide security system developers a good overview of relevant Common Criteria SFRs to address certain security aims. Compared to our work, Wu just focuses on the security requirements and does not describe the consequences of applying architectural tactics. We take benefit of security tactics which are a link between security quality attributes and architectural decisions. Linking the tactics to the Common Criteria SFRs allows us to extend Wu's connection between security quality attributes and SFRs to include the architectural decision which influences the quality attribute.

To further evaluate the effect of architectural decisions, such as the application of tactics, on system quality attributes, [Bass et al. 2003] suggest to construct scenarios and to evaluate different system architectures against these scenarios. Our work allows to establish a connection between the Common Criteria standard and these architecture evaluation methods by connecting SFRs to security tactics.

3. COMMON CRITERIA

The Common Criteria is an international security standard which evolved from security standards of the Canadian, French, German, Netherlands, UK and US government. The standard is used as a basis for the evaluation of security properties and allows to compare the security of IT systems [Common Criteria Recognition Arrangement 2009]. Part 1 of the standard gives a general overview of the certification process. Part 2 contains a collection of

Security Functional Requirements (SFRs) and Part 3 contains a collection of Security Assurance Requirements (SARs). Requirements are grouped into classes which are further refined into families and then further refined into components. The abbreviation of a requirement consists of three letters for the class, an underscore, three letters for the family, a dot and the component number. The requirement for cryptographic key generation (FCS_CKM.1), for example, is the first component of the *Cryptographic Key Management* family (CKM) which is member of the *Cryptographic Support* class (FCS).

For security certification of a target of evaluation (TOE), a Protection Profile has to be defined or reused. This profile is a general description of the type of TOE system and contains a set of SFRs and SARs against which the system has to be certified. If a Protection Profile already exists for the TOE domain, it can be reused. If it does not exist, it has to be constructed out of the SFRs and SARs given in the Common Criteria standard. This can be a tedious task due to the huge amount of requirements. To evaluate a TOE for security according to a Protection Profile, a Security Target has to be defined. This Security Target also consists of requirements for the TOE, but compared to the Protection Profile it is not general but bound to the specific implementation of the system. For the security target, a developer definitely has to create a set of Common Criteria requirements which the system has to meet. These requrements for Security Targets mostly consist of the Protection Profile requirements and probably additional requirements of the standard.

For the SARs several packages are defined which allow the developer to easily choose a suitable set of SARs. The packages are called Evaluation Assurance Levels (EALs) and represent the confidentiality one can have in the correct implementation of the system SFRs. There are, however, no packages defined for SFRs which makes it rather difficult for developers to choose an appropriate set of functional requirements. By mapping the SFRs to security tactics, this paper bridges this gap and allows developers to choose SFRs by deciding for security tactics.

4. GOAL STRUCTURING NOTATION

The Goal Structuring Notation (GSN) was developed by [Kelly and Weaver 2004] and is often used in the safety domain providing a structured way to argue for the achievement of specific goals. Recently, a standard for the GSN was published which contains the definitions of the notation and which presents approaches to use GSN in order to elaborate a specific goal [GSN Working Group 2011]. GSN was used to describe the rationale of safety tactics [Wu 2003] and several suggestions have been made to use GSN in the security domain [Kelly and Weaver 2004][Cockram and Lautieri 2007]. Figure 1 explains the GSN concepts which are used in this paper to link security tactics to Common Criteria SFRs.

5. SECURITY TACTICS

In this section we first give an overview of security tactics introduced by [Bass et al. 2003]. Furthermore, we cover the template we use to describe the security tactics and we present the whole refined and structured security tactic catalog.

5.1 Security Tactic Overview

Figure 2 gives an overview of the security tactics introduced by [Bass et al. 2003]. They are divided into three distinct categories. *Resisting attacks* covers security measures which can be applied in order to prevent attacks. These tactics address the confidentiality and integrity security attributes of a system. *Detecting Attacks* and *Recovering from an Attack* aim at handling successful attacks, where *Recovering from an Attack* focuses on availability issues of a system.

5.2 Tactic Template

In [Bass et al. 2003] each of the security tactics is just described in a single paragraph. We want to structure these tactics and add additional information. We do not use the common pattern description template consisting

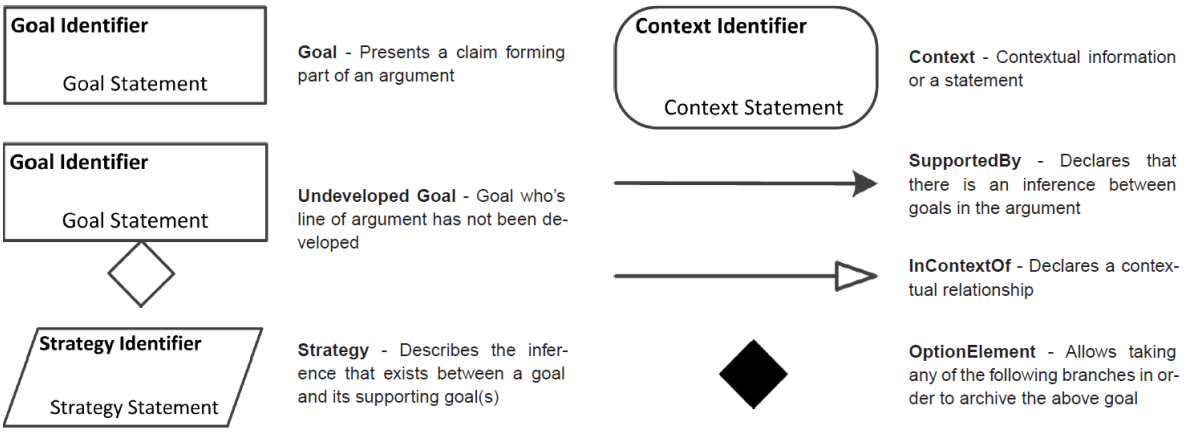


Fig. 1. GSN concepts used in this paper taken from [GSN Working Group 2011]

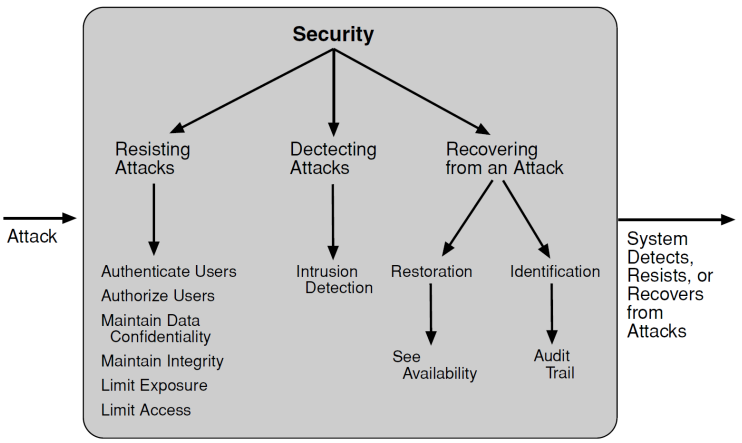


Fig. 2. Overview of security tactics [Bass et al. 2003]

of problem, forces, solution, and consequences to describe the security tactics. A tactic does not relate to a specific problem or context and tactics do not address trade-offs between forces [Ryoo et al. 2010]. We use the following sections to describe security tactics:

—NAME - The tactic name taken from [Bass et al. 2003]

—DESCRIPTION - A general description of the tactic based on [Bass et al. 2003]

—CONSEQUENCES - This tactic section describes consequences when applying the tactic. The consequences are partially taken from patterns presented in [Hafiz et al. 2011], [Schumacher et al. 2005], and [Kienzle et al. 2002] which apply the tactics.

—RELATED TACTICS - Gives information on related tactics. The information is partially based on [Kim et al. 2009].

—KNOWN USES - Gives examples for patterns applying the tactic. The examples are taken from the security pattern catalog presented in [Hafiz et al. 2011].

—COMMON CRITERIA RATIONALE - This tactic section is the main contribution of our work. Security tactics are linked to Common Criteria v.3.1 [Common Criteria Recognition Arrangement 2009] SFRs through GSN. GSN

allows us to use a structured approach to connect Common Criteria SFRs to architectural tactics. We do not break down the goal of the security tactic to a level of how it can be achieved. We break down the goal on subgoals required by the Common Criteria SFRs which can help achieving the overall goal of the security tactic.

We develop each tactic by gathering and structuring Common Criteria SFRs which are related to the tactic. The selection of Common Criteria SFRs used in the GSN is based on the following sources:

a) A thorough investigation of the Common Criteria standard Part 2
b) Protection Profiles
c) Wu's PhD thesis [Wu 2007a]

We found most of the relationships between tactics and SFRs by analyzing the SFR descriptions given in the Common Criteria standard. The SFR description of FDP_UIT (see footnote[1]), for example, suggests that this SFR is related to the *Maintain Integrity* tactic. We also analyzed approved Protection Profiles regarding their objectives and their related SFRs. The Separation Kernel Protection Profile [U.S. National Information Assurance Partnership 2007], for example, describes the objective *O.AUTHORIZED_SUBJECT*. This objective states that just authorized subjects are allowed to access restricted data. In the Protection Profile, the objective is reached if the FMT_MOF.1, FMT_MSA_EXP.1, FMT_MTD.1, and FMT_MCD_EXP.1 requirements are met. This gives us the hint that these SFRs are related to the *Authorize Users* tactic. Wu's PhD thesis [Wu 2007a] provides us with a link between the SFRs and security attributes such as privacy or confidentiality. Some of the security tactics (for example *Maintain Data Confidentiality*) are quite close to security attributes covered by Wu. The SFRs mapped to these security attributes, therefore, can directly be included in the collection of SFRs for the corresponding tactic.

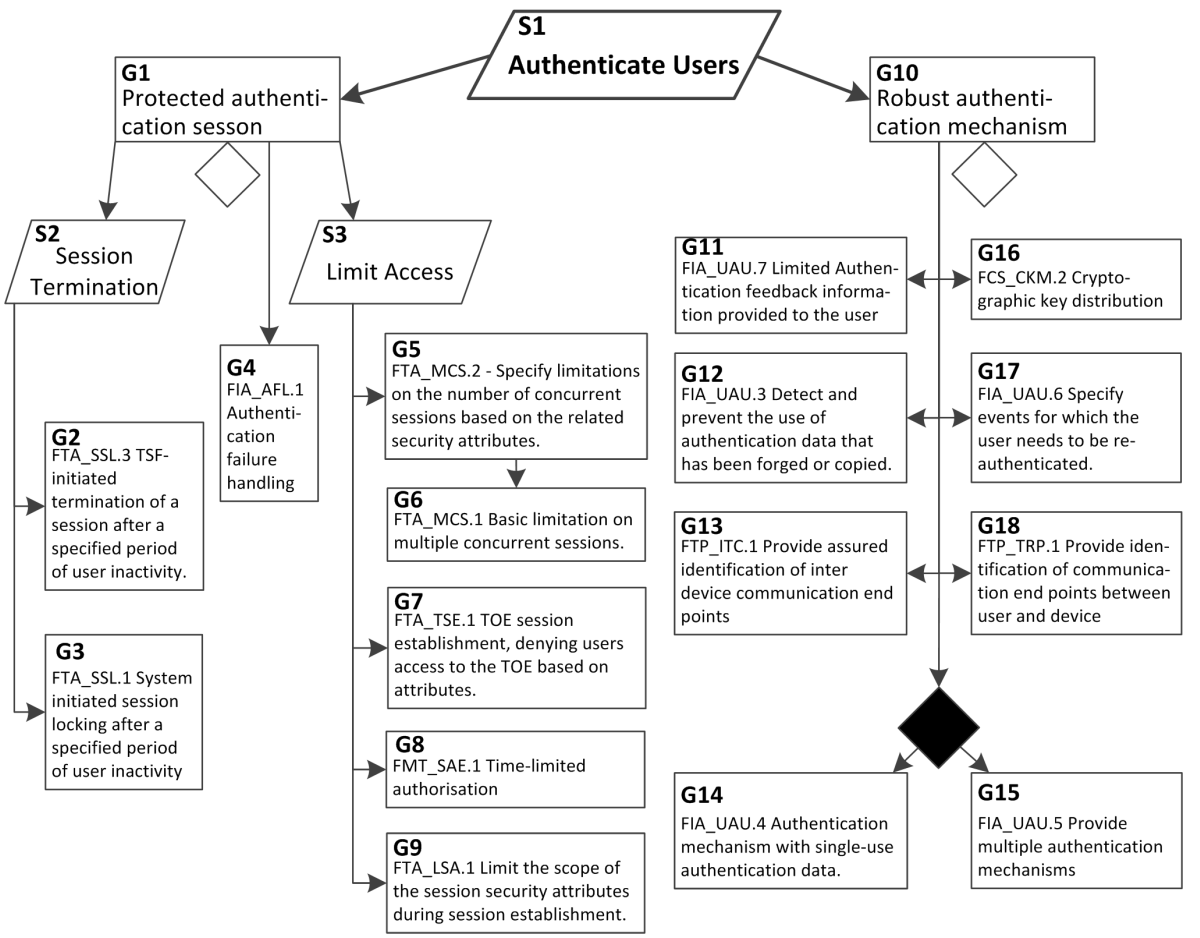
After collecting the SFRs for each tactic, we developed the GSN notation. For each tactic, we grouped SFRs with similar aims with respect to the tactic and related them to a more general goal or strategy. These general elements or the SFRs themselves are then connected to the security tactic. Some of the tactics are represented as strategies, some of them, however, are represented as goals in the GSN. This is, because some of the security tactics are not really design decisions, but more objectives (for example: *Maintain Data Confidentiality*).

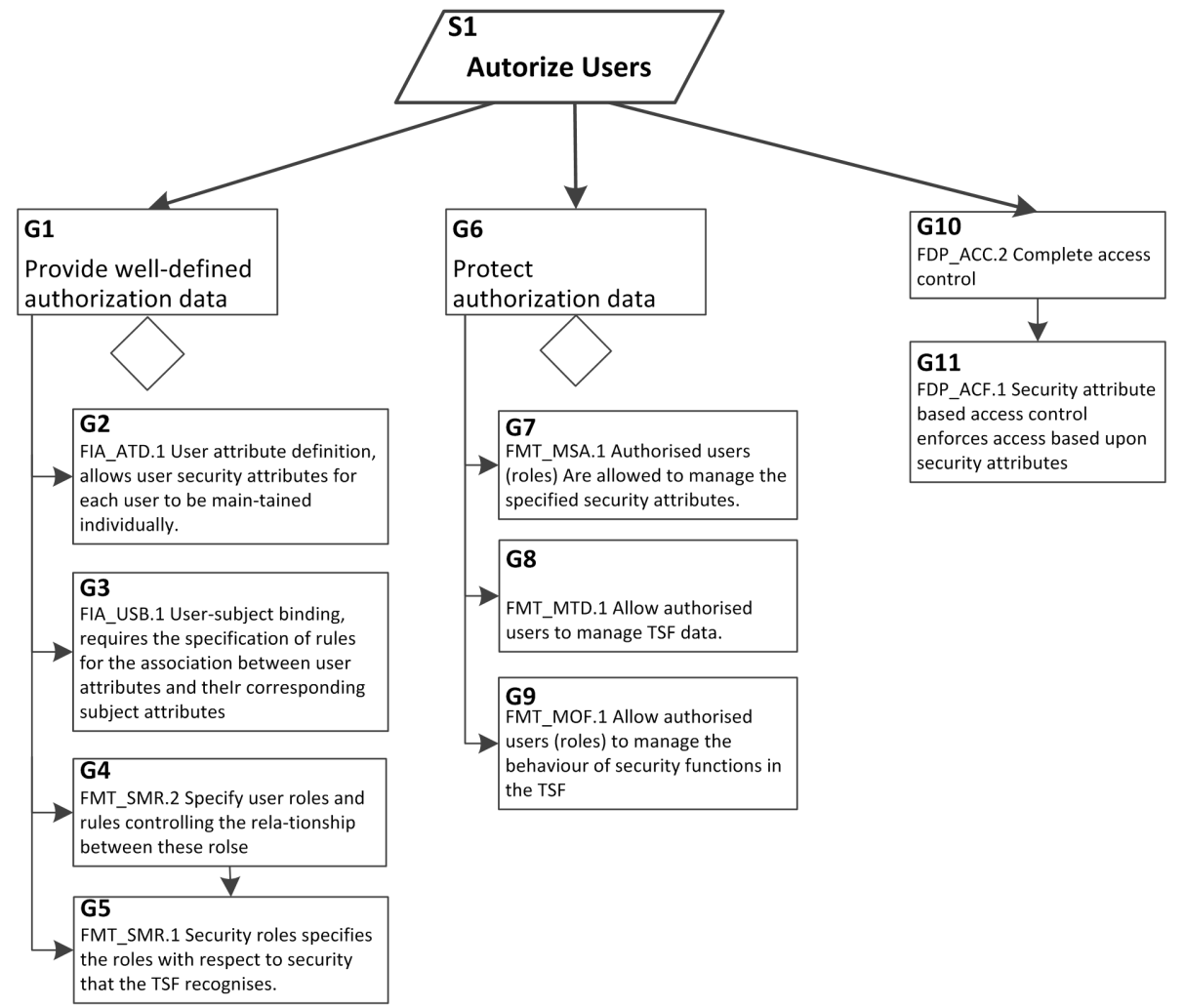
The completed GSN notation for the security tactics presents a set of requirements which allows a system architect to check which requirements have to be met in order to achieve either the stated goal, or which have to be met for applying the top-level strategy (the tactic) to a system architecture. An advantage of using GSN is that the tactics can be presented in a more structured way which allows to directly use the representation for architectural reasoning. GSNs provide a good basis for architectural reasoning regarding quality attributes and have already been successfully applied to the safety domain [Wu 2007b].

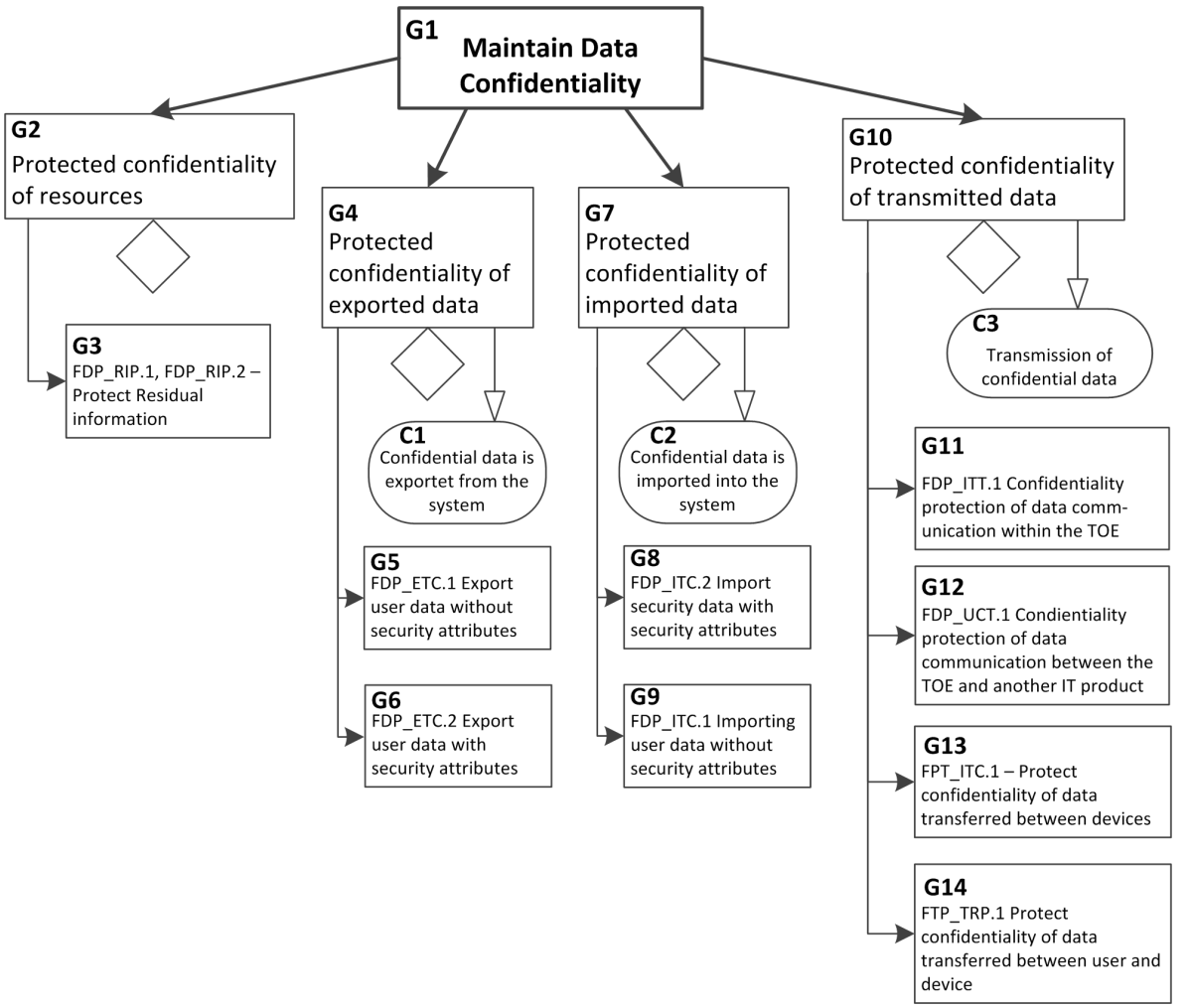
The GSN diagram allows to give a quick overview about the goals related to a security tactic. Additionally, the SFRs can provide more detailed information on how to achieve these goals.

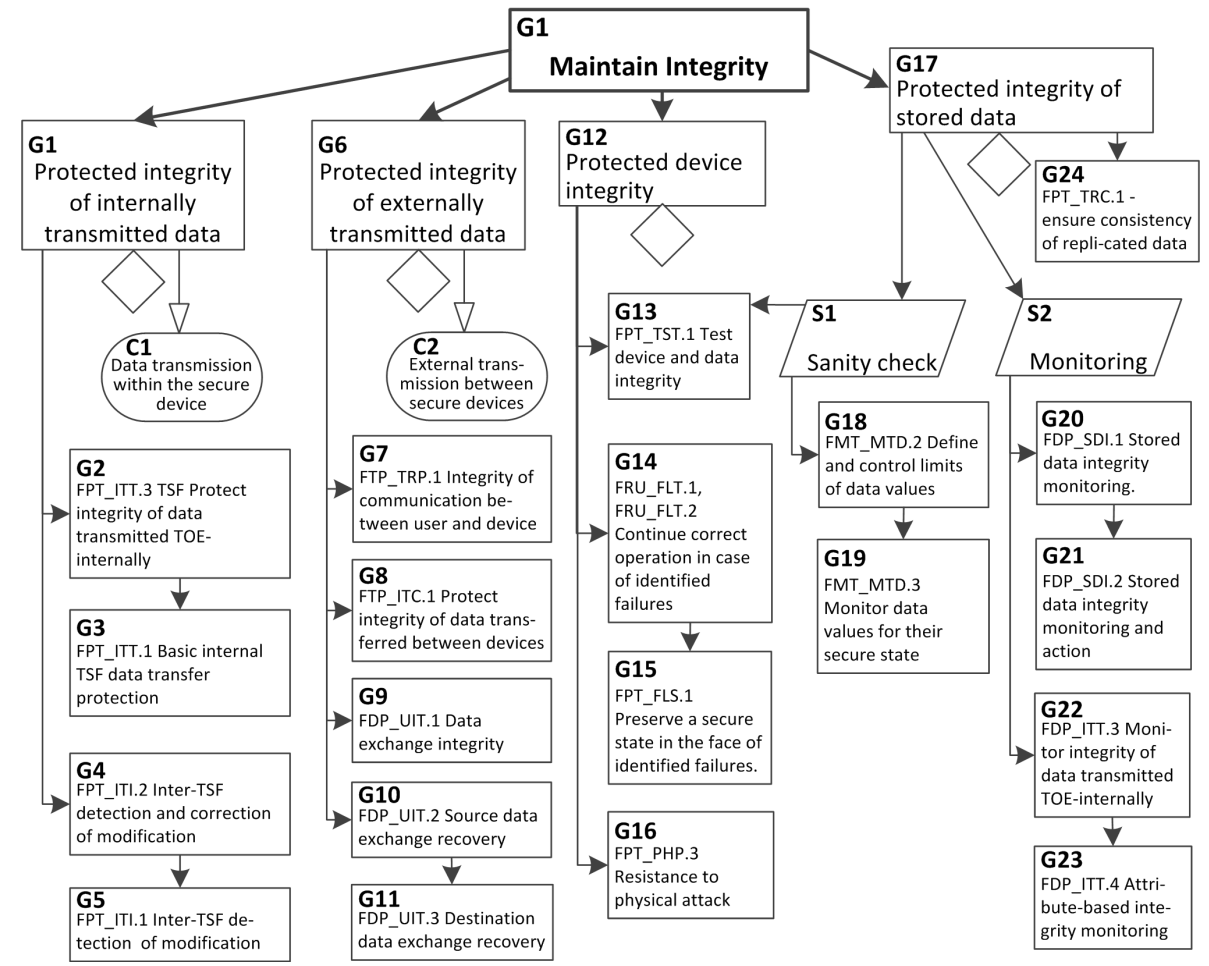
[1] FDP_UIT description: This family defines the requirements for providing integrity for user data in transit between the TOE and another trusted IT product and recovering from detectable errors. At a minimum, this family monitors the integrity of user data for modifications. Furthermore, this family supports different ways of correcting detected integrity errors. [Common Criteria Recognition Arrangement 2009]

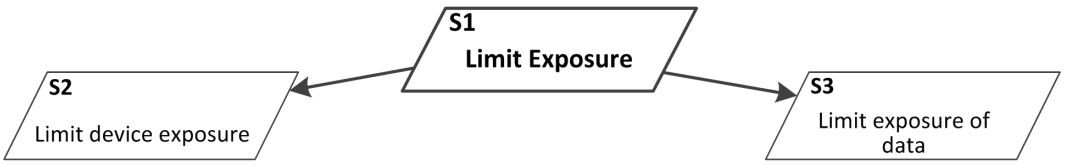
5.3 Tactics Catalog

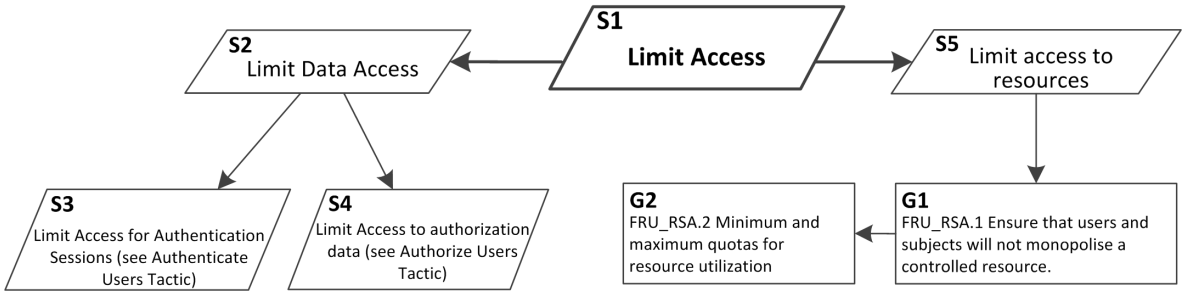
TACTIC NAME	**Authenticate Users**
DESCRIPTION	This tactic ensures that a user or computer is who he claims to be. Users/computers are required to own a secret which has to be established/distributed before authentication.
COMMON CRITERIA RATIONALE	 *Authenticate Users* is mainly based on the Common Criteria Identification and Authentication (FIA) class and on the TOE Access (FTA) class. FIA defines requirements for the authentication mechanism (G10) and FTA discusses how it can be protected (G1) by limiting the access (S3) and by maintaining session termination policies (S2).
CONSEQUENCES	Authentication mechanisms can make the access to a system more difficult and cumbersome. Authentication credentials have to be distributed/Maintained
RELATED TACTICS	*Authenticate Users* is supported by the *Limit Access* tactic (S3). *Authenticate Users* is often used in combination with *Authorize Users*
KNOWN USES	Account Lockout, Assertion Builder, Authentication Enforcer, Brokered Authentication, Message Intercepting Gateway

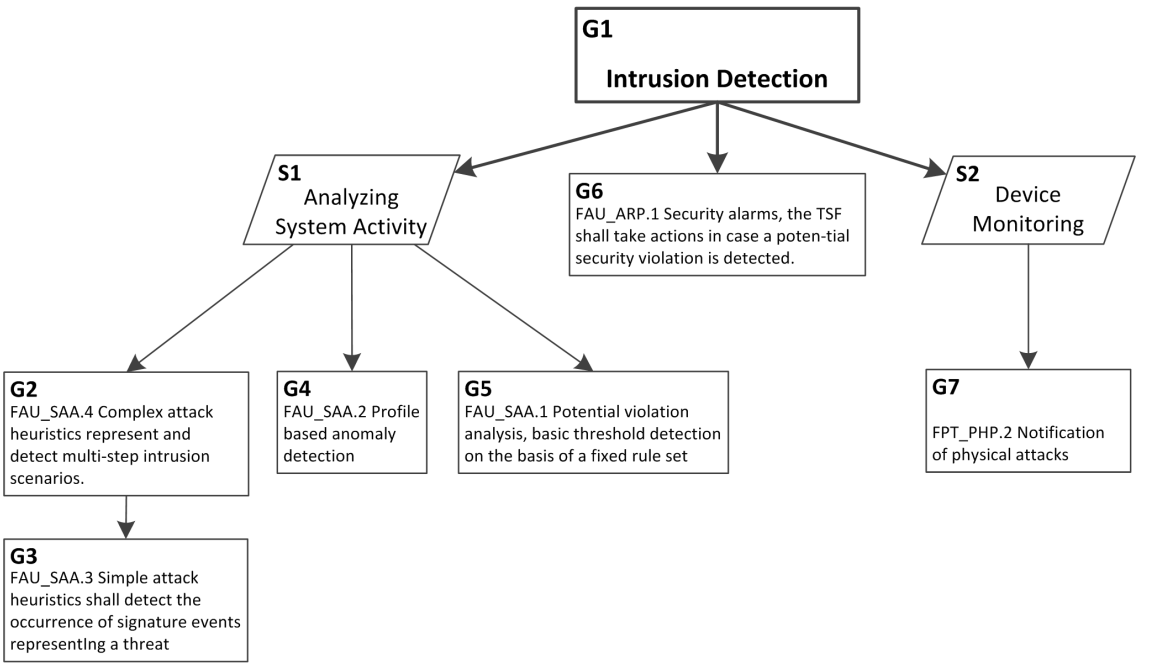
TACTIC NAME	**Authorize Users**
DESCRIPTION	This tactic ensures that only certain authenticated users have access to a resource
COMMON CRITERIA RATIONALE	 Common Criteria Requirements for the *Authenticate Users* tactic mainly cover specification (G1) and protection (G6) of authorization data
CONSEQUENCES	Rules regarding authorization can easily be changed, however, possibly many rules have to be maintained. Analyzing required rules for users and understanding their implications is a rather complex task [Schumacher et al. 2005].
RELATED TACTICS	The *Authenticate Users* tactic is required as a precondition for the *Authorize Users* tactic. Defining resources a user needs authorization for follows the *Limit Exposure* tactic
KNOWN USES	Assertion Builder, Brokered Authentication, Container Managed Security, Front Door, Intercepting Web Agent, Reference Monitor, Role Based Access Control, Secure Session Object, Security Context

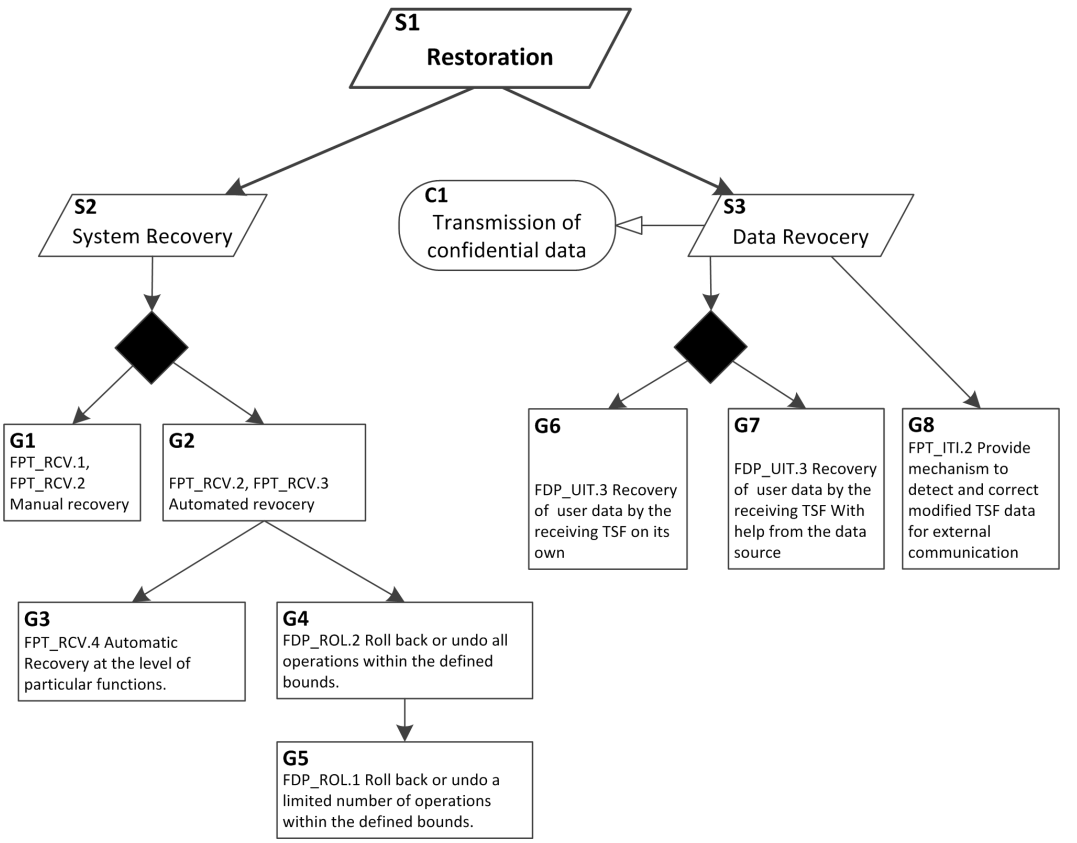
TACTIC NAME	**Maintain Data Confidentiality**
DESCRIPTION	Confidential data is protected from unauthorized access
COMMON CRITERIA RATIONALE	 Common Criteria Requirements for the *Maintain Data Confidentiality* tactic cover protection of stored (G2) and transmitted (G4) data as well as data imported (G12) into or exported (G9) from the system. *Maintain Data Confidentiality* is more a specific goal than a strategy of how to reach a goal and therefore is represented as a goal in the GSN
CONSEQUENCES	In order to enforce cryptographic protection of data, the required cryptographic secret has to be protected. Additional resources for computing and storing cryptographically protected data are required [Blakey and Heath 2004].
RELATED TACTICS	*Limit Exposure* helps to protect confidential data by eliminating possible attack vectors.
KNOWN USES	Encrypted Storage, Information Obscurity, Intercepting Web Agent, Secure Session Object

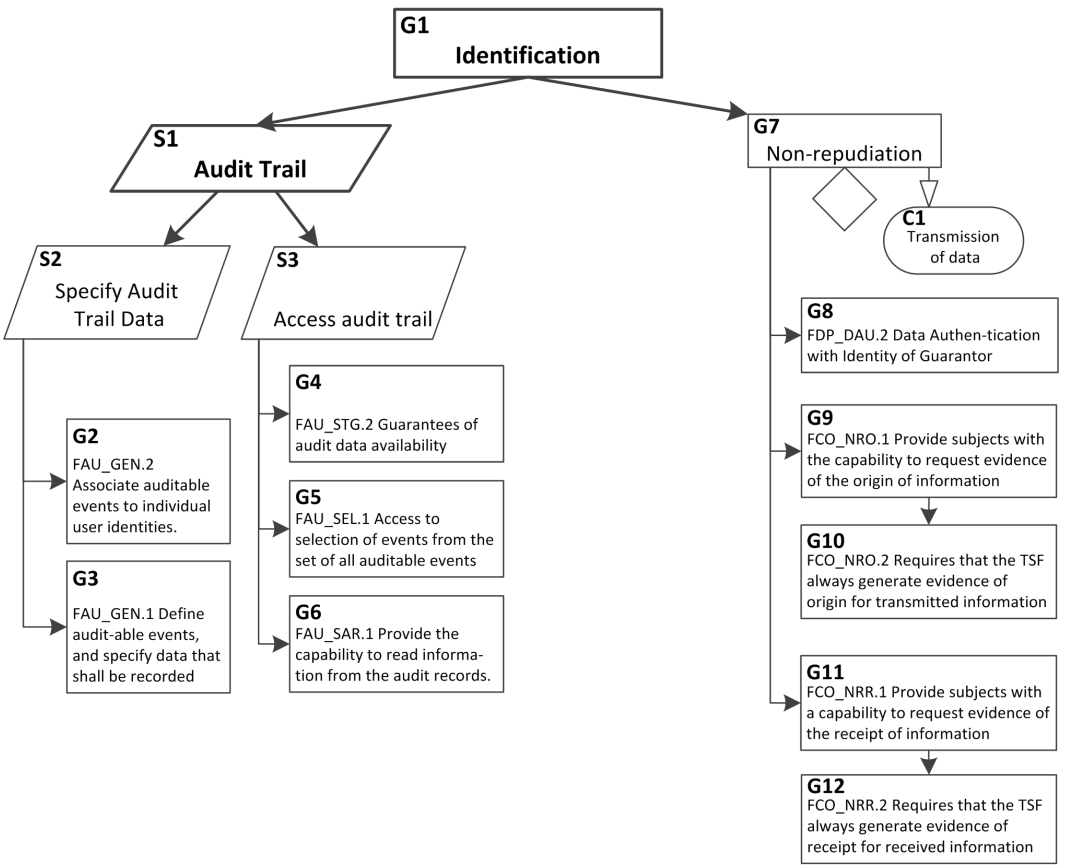
TACTIC NAME	**Maintain Integrity**
DESCRIPTION	Accidental and malicious system or data modifications should be prevented or detected
COMMON CRITERIA RATIONALE	 Most Common Criteria Requirements address the *Maintain Integrity* tactic. They can be divided in requirements protecting the device itself (G2), requirements protecting stored data (G7), and requirements protecting transmitted data (G15, G21). Most of the requirements come from the Common Criteria classes Protection of the TSF (FPT) and User Data Protection (FDP)
CONSEQUENCES	Additional resources are required to protect the integrity of data by means of cryptography [Blakey and Heath 2004]. Also any other integrity check measure requires additional resources in terms of redundant computation or storage.
RELATED TACTICS	For integrity protection the safety tactics *Sanity check* and *Monitoring* which are presented in [Wu 2003] are used. It is not surprising that safety tactics are used here, because integrity protection is part of safety measures as well [Avizienis et al. 2004].
KNOWN USES	Client Data Storage, Error Detection and Correction, Safe Data Structure

TACTIC NAME	**Limit Exposure**
DESCRIPTION	Possible attack vectors are decimated by limiting the ways security devices and data are accessible.
COMMON CRITERIA RATIONALE	 No common criteria requirements directly address *Limit Exposure*. However, if *Limit Exposure* is applied to a system, less Common Criteria requirements have to be used in order to achieve system security, because less threats affect the system.
CONSEQUENCES	*Limit Exposure* is highly desirable for a system, however, it may not always be possible to enforce this tactic. *Limit Exposure* decimates possible attack vectors and therefore makes security validation easier. The tactic may however decrease system functionality.
RELATED TACTICS	*Limit Access* is a way to enforce *Limit Exposure*
KNOWN USES	Compartmentalization, Hidden Implementation, Trust Partitioning

TACTIC NAME	**Limit Access**
DESCRIPTION	Access to a resource is disabled. The user does not even have the possibility to access it [Schumacher et al. 2005].
COMMON CRITERIA RATIONALE	 Common Criteria provides requirements limiting the access to data (S2) and to system resources like memory usage or processing power (S5).
CONSEQUENCES	*Limit Access* decimates possible attack vectors. However, system functionality can be affected. Administrators do not have to define access rights and enforce access rules, because access is completely denied [Schumacher et al. 2005].
RELATED TACTICS	*Limit Access* is a form of *Limit Exposure*. *Limit Access* is used for *Authenticate Users*. Also the *Authorize Users* and *Maintain Data Confidentiality* tactic inherently use *Limit Access*
KNOWN USES	Authentication Enforcer, Chroot Jail, Demilitarized Zone, Front Door, Message Intercepting Gateway, Packet Filter Firewall, Policy Enforcement Point

TACTIC NAME	**Intrusion Detection**
DESCRIPTION	Detect ongoing or past attacks on the system
COMMON CRITERIA RATIONALE	 Most *Intrusion Detection* relevant Common Criteria Requirements are found in the Security Audit (FAU) class, who's description already says that the class can be taken for intrusion detection requirements.
CONSEQUENCES	Measures allowing to identify attacks and measures taken when an attack is detected have to be specified. *Intrusion Detection* requires additional resources to log or monitor relevant data.
RELATED TACTICS	-
KNOWN USES	Dynamic Service Management

TACTIC NAME	**Restoration - Availability Tactics**
DESCRIPTION	The system can be restored after an attack
COMMON CRITERIA RATIONALE	 The Common Criteria requirements cover system and data recovery with the classes Protection of the TSF (FPT) and User Data Protection (FDP)
CONSEQUENCES	Compared to simple recovery for availability, in the security domain special care has to be taken when maintaining copies of the system for *Restoration*, because this includes that multiple copies of the system can be attacked [Im and McGregor 2007].
RELATED TACTICS	*Restoration* can be in conflict with *Maintain Data Confidentiality* if multiple copies of a system have to be maintained. *Limit Access* can be applied in that case to not increase the attack surface. *Restoration* is an availability tactics and any recovery tactic presented in [Bass et al. 2003] to meet availability aims can be applied.
KNOWN USES	Checkpointed System, Error Detection and Correction, Replicated System, Standby, Tandem System

TACTIC NAME	**Identification - Audit Trail**
DESCRIPTION	Actions performed by a user are logged including an identity link
COMMON CRITERIA RATIONALE	 Most requirements come from the Common Criteria FAU class which explicitly addresses security audits. Other requirements come from the Communication (FCO) class and cover non-repudiation (G7) of sent or received messages. Non-repudiation is not handled as a separate security tactics in [Bass et al. 2003], but can be seen as part of the general *Identification* goal (G1).
CONSEQUENCES	Additional security relevant data for the audit trail has to be stored and protected. *Identification* possibly conflicts with user privacy requirements [Schumacher et al. 2005].
RELATED TACTICS	Maintaining an additional record of confidential audit data works against the *Limit Exposure* tactic. *Authenticate Users*, *Authorize Users*, *Maintain Data Confidentiality*, and *Maintain Integrity* are necessary to protect the audit trail.
KNOWN USES	Audit Interception, Secure Logger

Security Tactics / Common Criteria SFR Classes	Authenticate Users	Authorize Users	Maintain Data Confidentiality	Maintain Integrity	Limit Exposure	Limit Access	Intrusion Detection	Restoration Availability	Identification Audit Trail
FAU (Security Audit)							X		X
FCO (Communication)									X
FCS (Cryptographic Support)	X								
FDP (User Data Protection)	X	X	X	X				X	X
FIA (Identification&Authentication)	X	X							
FMT (Security Management)	X	X		X					
FPR (Privacy)									
FPT (Protection of the TSF)			X	X			X	X	
FRU (Resource Utilisation)				X		X			
FTA (TOE Access)	X								
FTP (Trusted Path/Channels)	X		X	X					

Table I: Mapping of the Common Criteria SFR classes to security tactics

6. DISCUSSION

Table I shows that no Common Criteria SFRs could be found which directly address the *Limit exposure* tactic. However, *Limit exposure* is a valid security tactic. It is a basic security principle and is often applied to security patterns. *Limit Exposure* fulfills all necessary requirements for tactics (atomicity, force limitation, problem unspecific, completeness, no forces trade-off) which are described in [Ryoo et al. 2010]. However, *Limit Exposure* does not have a direct functional aim and therefore is not directly addressed by the SFRs. This illustrates that the SFRs can be taken to enhance the tactics, but they cannot be taken as the single basis for security tactics.

The security tactics also do not address all aspects of security. Privacy, for example is not handled at all. The whole Common Criteria class addressing privacy (FPR) is not mapped to any of the security tactics. Apart from this class, at least parts of all other SFR classes were mapped to at least one security tactic. This shows us that the security tactics appear to be incomplete, because they do not address all security quality attributes.

Most of the SFRs are mapped to the security tactics regarding authorization, authentication, confidentiality, and integrity. This inidcates that these tactics are especially suited to be further refined into sub-tactics, which could be based on the presented strategies in the goal structuring notation of the corresponding tactic. Confidentiality, for example, can be seen as a separate quality attribute but it is just addressed by the *Maintain Data Confidentiality* tactic. Also the integrity quality attribute is just addressed by one security tactic, *Maintain Integrity*. These two tactics are mapped to many SFRs which suggests that these two security tactics should be further refined. Another explanation for the imbalanced SFR distribution is that some security tactics rather address system requirements than architectural design decisions. *Maintain integrity*, for example just says that the quality attribute integrity has to be met and gives no design decision on how to achieve that. This is the reason why we represented the *Maintain Integrity* tactic as a goal in the GSN. This indicates that the tactic is not very well chosen. Another indicator for this is that well known security principles such as defense in depth or the least privilege principle cannot be found in the security tactics catalog. Therefore, we think that the security tactics catalog is rather incomplete and we leave it up to future work to revisit security tactics regarding their structure and completeness.

7. CONCLUSIONS

In this paper we present a full catalog of security tactics with focus on the link between the tactics and the Common Criteria security standard. Moreover, this catalog provides more detailed and structured descriptions of security tactics compared to the initially presented tactics by [Bass et al. 2003].

The link between security tactics and SFRs allows easier evaluation of the influence of SFRs on system architectures. To evaluate a system architecture, popular software architecture evaluation methods such as ATAM [Kazman et al. 2000] or CBAM [Kazman and Klein 2002] use scenarios to compare similar architectures differing in their architectural styles which can be composed of tactics [Kim et al. 2009]. The link of security tactics to the SFRs provides an initial step to enable reasoning with these architecture evaluation methods about the influence of SFRs on system quality attributes.

Another advantage of the refined security tactics catalog is that system developers who want to apply a security tactic, just have to look at the SFRs connected to these tactics to get a basic idea how the tactic can be implemented. For systems which have to be Common Criteria certified, the refined catalog allows a system architect to get on overview of the SFRs related to a tactic. This can reduce the certification effort by giving an initial advice on the SFRs which should be included for the system certification. The other way around, if the SFRs for a system are already defined (e.g. by a Protection Profile), then the system architects can use the catalog to easily see which tactics can be chosen in order to fulfill the given SFRs.

Another application of our refined security tactics catalog could be to enhance the information of existing security patterns which use the tactics. [Kumar and Prabhakar 2010] explains how patterns can be decomposed into its basic underlying tactics. By applying this approach to security patterns, the information about the consequences for security patterns could be extended by the effect of the underlying tactics on security certification.

We believe that the presented collection of security tactics encourages the usage of security tactics and security patterns for products which have to be certified according to the Common Criteria standard.

ACKNOWLEDGMENTS

We would like to thank our shepherd Kiran Kumar very much for significantly improving the structure and the content of this paper by giving helpful hints and comments during the shepherding process.

REFERENCES

AL-DAAJEH, S. H., AL-QTAISH, R. E., AND AL-QIREM, F. 2011. Engineering Dependability to Embedded Systems Software via Tactics. *International Journal of Software Engineering and Its Application 5,* 4, 45–62.

AVIZIENIS, A., LAPRIE, J.-C., RANDELL, B., AND LANDWEHR, C. 2004. Basic concepts and taxonomy of dependable and secure computing. *IEEE Transactions on Dependable and Secure Computing 1,* 1, 11–33.

BASS, L., CLEMENTS, P., AND KAZMAN, R. 2003. *Software Architecture in Practice* 2nd Ed. Addison-Wesley, Reading, Massachusetts.

BIALAS, A. 2011a. Common Criteria Related Security Design Patterns for Intelligent Sensors - Knowledge Engineering-Based Implementation. *Sensors (Basel, Switzerland) 11,* 8, 85–114.

BIALAS, A. 2011b. Patterns Improving the Common Criteria Compliant IT Security Development Process. In *Dependable Computer Systems (Advances in Intelligent and Soft Computing)* Volume 97 Ed. Springer-Verlag, 1–16.

BLAKEY, B. AND HEATH, C. 2004. Security Design Patterns. Tech. rep., The Open Group.

COCKRAM, T. J. AND LAUTIERI, S. R. 2007. Combining Security and Safety Principle in Practice. In *2nd Institution of Engineering and Technology International Conference on System Safety*. IEEE, 159–164.

COMMON CRITERIA RECOGNITION ARRANGEMENT. 2009. Common Criteria Standard v3.1. http://www.commoncriteriaportal.org/cc/.

GSN WORKING GROUP. 2011. GSN Community Standard Version 1. http://www.goalstructuringnotation.info/.

HAFIZ, M., ADAMCZYK, P., AND JOHNSON, R. 2011. Growing a Pattern Language (for Security). In *18th Conference on Pattern Languages of Programs*.

HOUMB, S. H., ISLAM, S., KNAUSS, E., JÜRJENS, J., AND SCHNEIDER, K. 2009. Eliciting security requirements and tracing them to design: an integration of Common Criteria, heuristics, and UMLsec. *Requirements Engineering 15,* 1, 63–93.

IM, T. AND MCGREGOR, J. 2007. Security in the Context of Dependability. In *Cyber Security and Information Infrastructure Workshop*.

KAZMAN, R. AND KLEIN, M. 2002. Design Decisions : An Economic Approach. Tech. Rep. September, Software Engineering Institute Carnegie Mellon University.

KAZMAN, R., KLEIN, M., AND CLEMENTS, P. 2000. ATAM : Method for Architecture Evaluation. Tech. Rep. August, Software Engineering Institute Carnegie Mellon University.

KELLY, T. AND WEAVER, R. 2004. The Goal Structuring Notation Ű A Safety Argument Notation. In *Proceedings of the Dependable Systems and Networks Conference*.

KIENZLE, D. M., ELDER, M. C., TYREE, D., AND EDARDS-HEWITT, J. 2002. Security Patterns Repository. Tech. rep.

KIM, S., KIM, D.-K., LU, L., AND PARK, S. 2009. Quality-driven architecture development using architectural tactics. *Journal of Systems and Software 82,* 8, 1211–1231.

KUMAR, K. AND PRABHAKAR, T. V. 2010. Pattern-oriented Knowledge Model for Architecture Design. In *17th Conference on Pattern Languages of Programs*.

RYOO, J., LAPLANTE, P., AND KAZMAN, R. 2010. A Methodology for Mining Security Tactics from Security Patterns. In *2010 43rd Hawaii International Conference on System Sciences*. IEEE, 1–5.

SCHUMACHER, M. 2002. Security Patterns and Security Standards. In *Proceedings of the 7th European Conference on Pattern Languages of Programs*.

SCHUMACHER, M. 2003. *Security Engineering with Patterns*. Springer, Heidelberg, Germany.

SCHUMACHER, M., FERNANDEZ, E. B., HYBERTSON, D., BUSCHMANN, F., AND SOMMERLAD, P. 2005. *Security Patterns - Integrating Security and Systems Engineering*. John Wiley & Sons.

SCOTT, J. AND KAZMAN, R. 2009. Realizing and Refining Architectural Tactics : Availability. Tech. Rep. August, Carnegie Mellon Software Engineering Institute.

U.S. NATIONAL INFORMATION ASSURANCE PARTNERSHIP. 2007. U.S. Government Protection Profile for Separation Kernels in Environments Requiring High Robustness.

WU, D. 2007a. Security functional requirements analysis for developing secure software. Ph.D. thesis, University of Southern California.

WU, W. 2003. Safety Tactics for Software Architecture Design. M.S. thesis, The University of York.

WU, W. 2007b. Architectural reasoning for safety- critical software applications. Ph.D. thesis, University of York.

WYETH, A. M. 2009. Formal Specification of Software Architecture Design Tactics for the Security Quality Attribute. M.S. thesis, California State University.