Fraunhofer
IESE

Security Indicators – A State of the Art Survey

Public Report

Authors:
Manuel Rudolph
Dr. Reinhard Schwarz

IESE-Report No. 043.12/E
Version 1.0
March 2012

A publication by Fraunhofer IESE

Fraunhofer IESE is an institute of the Fraunhofer Gesellschaft.
The institute transfers innovative software development techniques, methods and tools into industrial practice, assists companies in building software competencies customized to their needs, and helps them to establish a competitive market position.

Fraunhofer IESE is directed by
Prof. Dr. Dieter Rombach (Executive Director)
Prof. Dr. Peter Liggesmeyer (Director)
Fraunhofer-Platz 1
67663 Kaiserslautern

Abstract

Measurement is one of the foundations of sound engineering practices, because—as Tom DeMarco put it—you cannot control what you can't measure. This principle should also apply to software security engineering. However, providing useful metrics or at least indicators for characterizing the security properties of a software system is surprisingly challenging.

The research community is well aware of the urgent need for security metrics, and it has put significant research effort into this field. Numerous qualitative and quantitative security measures have been proposed in the scientific literature, but few of them found wide-spread adoption by practitioners. Due to the significant body of work, it has become increasingly difficult to overlook the state of the art in specifying, determining, comparing, or predicting security qualities.

This report surveys the published work on security indicators. In the context of this survey, a security indicator is understood as an observable characteristic that correlates with a desired security property. Our survey covers current research into qualitative and quantitative security indicators as well as applied key performance indicators and security standards.

We developed a uniform classification scheme for categorizing and comparing the indicators that we elicited. Based on this classification, our survey reveals trends and deficiencies in security research and security practice. It also suggests explanations for the apparent difficulties in providing meaningful security indicators. Moreover, our classification can guide practitioners to adequate methods for the specification of security requirements and for the measurement of relevant security attributes of their products and processes.

Keywords: software security, security measure, security metrics, security indicator, security improvement, classification model, classification tree

Table of Contents

1 Introduction

> *»In physical science the first essential step in the direction of learning any subject is to find principles of numerical reckoning and practicable methods for measuring some quality connected with it. I often say that when you can measure what you are speaking about, and express it in numbers, you know something about it; but when you cannot measure it, when you cannot express it in numbers, your knowledge is of a meager and unsatisfactory kind; it may be the beginning of knowledge, but you have scarcely in your thoughts advanced to the state of science, whatever the matter may be«.*
>
> (Lord Kelvin)

> *»Not everything that counts can be counted, and not everything that can be counted counts«.*
>
> (Albert Einstein)

The importance of information security for economy, governments and our private lives rises rapidly as more and more security breaches occur and are made public. Nowadays, countless incidents flood the news about fatal security breaches in, for example, enterprises' information systems, where millions of sensitive personal data records get stolen or lost. Malfunctions or missing security controls in software, hardware, networks or whole systems allow malicious intruders to abuse information systems and to steal, manipulate, or destroy sensitive data.

For example, in January 2009 a successful attack compromised approximately 130 Million credit cards at the fifth largest credit card processor Heartland Payment Systems [OSF10a]. Malicious code was planted on the company's payment processing network by a group of hackers and remained undetected for an unknown period of time [Kre09, Vor09].

In July 2009 about 40 websites of the governments, newspapers and merchants in the United States and South Korea fell victim to concentrated Distributed Denial-of-Service attacks and were not accessible. The initiator could not be identified conclusively, but it was rumored that the North Korean government was behind these attacks [Sud09, Die09].

Depending on the cause of the security issue, the consequences for the aggrieved party can be massive tangible and intangible losses, e.g. losses in reputation, decreasing business volumes or high costs for repairing any damage.

But IT-security is not only a technical, but also an organizational issue. In fact, there is often a lack of security in the organization of enterprises, e.g. a missing

enterprise-wide security policy, which can lead to social attacks or accidental disclosure of confidential data.

For example, in March 2010 a stolen portable media device from ECMC Group Inc. caused the loss of 3.3 million personal data records including names, addresses, birth dates and Social Security numbers. The device was not sufficiently physically protected against theft [OSF10, Pil10]. A similar incident occurred in 2008 when the U.S. National Archives and Records Administration improperly disposed a hard drive that contained approximately 76 million records of veterans, also including Social Security numbers [Sin09].

There are hundreds of such incidents in which data was somehow stolen or accidentally disclosed. In most cases the damage would have been avoidable by using adequate security precautions.

To prevent or at least minimize possibilities for such security incidents, enterprises, organizations, and governments must become more sensitive to security challenges, and they need information systems that are less vulnerable to security assaults. To this end, developers must work out more robust and secure system designs, implement them compliantly and use mature and reproducible security engineering processes to guarantee an adequate level of security in their systems. To better control the security properties of information technology and information processing, both developers and operators need advanced methods to specify and to control security in IT systems and IT processes.

Unfortunately the security research lacks behind other (software) engineering disciplines.

> *»In evolutionary terms, the information security field is more than a decade behind software development. By that, I mean that we haven't had a single meaningful change in security architecture in 13 years«.*
>
> (Gunnar Peterson, [Pet09])

A fundamental problem is the difficult and not yet mastered task of measuring and proving security. There are only few significant security indicators or quantitative security metrics available to characterize the security of software or software systems. However, adequate indicators and metrics to prescribe, measure, predict, and prove security properties are the key to controlling security and privacy during conception, development, and operation of IT systems.

1.1 Purpose of this Survey

This report surveys the current state of the art in the field of security indicators. In the context of this survey, a security indicator is understood as an observable characteristic that correlates with a desired security property. Indicators can be determined, for example, by using security metrics and quantifying existing characteristics, or by estimating the rate of compliance of the security target with predefined criteria. Therefore, our survey covers current research into

qualitative and quantitative security indicators as well as applied key performance indicators and security standards.

All indicators are classified using a uniform classification model. The classification shall reveal trends and deficiencies in security research and security practice. In addition, the report can guide developers to adequate methods for the specification of security requirements and for the measurement of security in their products and processes. A classification tree shall structure and group all security indicator approaches. It should provide suggestions for applicable security indicators for a given scenario.

1.2 Structure of the Report

Chapter 2 defines the key terminology used in this report to characterize security indicator approaches.

Chapter 3 presents a classification model for determining the state of the art in the domain of security indicators. This model lists indicator- or metric-based security approaches in tabular overviews, describing their characteristics and categorizing them into functionality classes. The classification helps to work out their purpose, application domain, and other characteristics.

As many publications contain more than one indicator, we concentrate in Chapter 4 on analyzing the individual measures and metrics that are proposed in the surveyed publications. We map single measures and metrics to measureable security properties. After that we categorize all measureable security properties into five main branches in a tree-like structure. These categories help to determine the maturity of the proposed security indicators and their coverage of relevant security properties.

In Chapter 5 we discuss the strength and weaknesses of know security indicators. We scrutinize their maturity, and our discussion reveals important fields for further study.

All approaches that are listed in our classification and the classification tree of the security indicators are described in more detail in the internal report [RS12].

1.3 Related Work

To the best of our knowledge, no comprehensive classification scheme for security specifying and measuring methods is currently available. The classification used in this report was inspired by a similar scheme for the classification of general software quality models: Quamoco — Map of Existing Quality Models [EHM+09]. Our scheme assigns similar attributes to characterize the methods under consideration, but its focus is on a single quality attribute, security.

In [Ver09] Verendel provides a survey of the state-of-the art in quantifying security. However, his survey is restricted to the area of operational security, exclud-

ing aspects such as secure software development. He classified a significant part of work in the field between 1981 and 2008 with respect to several properties. Among these properties are the perspective (i.e., »the conceptual viewpoint from which the approach to security is taken« [Ver09]), security related assumptions, and the kind of validation used to evaluate the work. Verendel concludes that most methods are not validated in quantifying or measuring operational security and that it is risky to rely on unevaluated methods.

Vaughn, Henning and Siraj introduce a taxonomy for security metrics [VHS03]. They divide security metrics in two groups: Metrics for Organizational Security and Metrics for Technical Targets of Assessment. In these groups they define several categories that characterize the analyzed metrics, such as Process Maturity Metrics, Effectiveness Metrics, or Survivability Metrics. However, [VHS03] does not apply the proposed classification scheme to provide a systematic catalogue of existing work on security metrics.

In [VFP04] Villarrubia, Fernández-Medina, and Piattini present a classification model for security metrics. They characterize security metrics according to application details, purpose, benefit, information return and other features that the metrics provide. Unfortunately, they have only selected security metrics that fit in their classification scheme and that have been described precisely enough to determine the classification attribute values.

Boyer and McQueen define seven security ideals in [BM07]. They assert that generally accepted security ideals can be mapped to their definition. In addition they attach known security metrics to their security ideals in order to propose a methodology for measuring the effectiveness of the application of the security ideals.

Stoddard et al. published a brief overview of existing security metrics systems, standards and scoring tools that have the potential to be applied to process control systems in the oil and gas industry [SH05]. In addition, they present risk assessment techniques, risk filtering and ranking metrics and ongoing efforts for developing new metrics tools specifically designed for the evaluation of process control system security.

1.4 Our Approach to Reviewing the Literature

For our literature review we used the following strategy consisting of two essential steps. In the first step we collected publications that are related to our topic in the broadest sense. We used the four sources listed below to collect a broad spectrum of works:

- Known publications: We evaluated relevant papers, books and standards that we had discovered prior to this literature review. In addition, we read several surveys on related subjects such as [GWM+07, Ver09] to identify promising approaches that we had not been aware of before.

- Known indicators: We searched for publications covering security indicators that we already knew using the appropriate search engines Google Scholar (http://scholar.google.de) and IEEE Xplore (http://ieeexplore.ieee.org).
- Keyword Search for new types of indicators: We used the search engines mentioned above to perform a keyword search using (among others) the following keywords: "Security AND Indicator", "Security AND Metric", "Security AND Aspect", "Software AND Security AND Checklist", "Security AND Process".
- Publication References: We used the reference lists of known publications to find more related work. Especially other surveys of security measurement were explored.

We ended our literature search after not finding any more new and promising publications that are related to our topic. In addition, the high coverage of publications that we found by those that have been listed in other security measurement surveys let us conclude our search.

In the second step we filtered out publications that have only little relation to our topic or that are in an early stage of development and not yet applicable. After that we chose, to the best of our knowledge, the most promising and mature work on security indicators.

2 Terminology

In this chapter we define terminology as it is understood in the context of this survey.

Definition: **Security Indicator**

A security indicator is any observable characteristic that correlates (or is assumed to correlate) with a desired security property.

Examples for indicators are measures, metrics, or the rate of compliance with security criteria catalogs and best practices. The set of feasible indicator values is assumed to form (at least) a nominal scale.

Note that for many proposed indicators the required correlation with security has not been formally established, but is only postulated based on informal reasoning.

Definition: **Security Measure**

A security measure assigns to each measured object a security indicator value from an ordinal scale according to a well-defined measurement procedure.

In many cases, the measured values are numbers, but measures may also assign non-numeric designators such as { LOW, MEDIUM, HIGH }.

Definition: **Security Metric**

Basically we understand a security metric as Mary Ann Davidson describes it in her characterization [Dav09]:

»A good security metric should:

- *Motivate good/correct behavior (not promote evasive tactics just to make the numbers look good).*
- *Prompt additional questions ("Why? How?") to understand what is influencing the numbers.*
- *Answer basic questions of goodness (e.g., "Are we doing better or worse?").*
- *Be objective and measureable, even if correlation may not equal causality.«*

In addition to Davidson's characterization, in our understanding a security metric is a security measure satisfying the following additional requirements:

- Clarity of Scope: It must be defined exactly which security feature the metric characterizes, and which it does not.

- Foundation: Measurement must be based on a security model that provides a hypothesis about the relation between the measured value and actual security properties of the target of measurement; this model must allow validation or falsification.
- Reproducibility: A measurement with the same metric applied on the same object must yield the same result, independent from the particular assessor.
- Relevance: A metric must reveal useful information about an object that can be used for decision making and provide necessary details for system specification, system comparison, or for the prediction of system properties.
- Well-defined Parameters. The origin and the determination of parameters of a metric calculation must be defined clearly.
- Well-defined Scale: The scale of a metric must be defined clearly, including the type of scale, the range of values and their interpretation.
- Established Baseline: A metric must specify a baseline for benchmarking. This baseline provides values (or gives advice how to define values) that represent the optimal result of the metric under given circumstances.

3 Classification of Know Approaches

In this Chapter, we introduce a classification model to structure the results of our literature review. The focus of our classification model is on the purpose, application scope, and usage of the surveyed approaches in the system life cycle. This classification is meant to be a preparatory work for a grouping of the security indicators proposed in the literature. It shall provide a first estimation of the state of the art in the field of security specification and measurement.

Our classification model was inspired by the *Quamoco Map of Existing Quality Models* [EHM+09]; it was developed before the literature research started.

3.1 Attribute Description

This section defines the attributes of the classification model and it describes the values they can take.

Attribute: **Name**

This attribute specifies the name of the described work. If the available sources do not explicitly name the security indicators that are proposed, a short descriptive phrase is provided.

Attribute: **Target**

This attribute denotes the target on which the method and its proposed indicator(s) can be applied. More than one value may be chosen.

Attribute values:

- Product: The indicator refers to the security of the target, e.g., software products or parts of them (source code, components), networks, systems.
- Process: The indicator refers to security-related parts of a process, e.g., a development process, or a maintenance process. The underlying rationale is that improving the process will indirectly improve the security of the product.
- Resources: The indicator refers to security-related attributes of resources that are used by a product or process. Resources can be, for example, supporting tools, involved persons or other material used in production.

Attribute: **Focus**

This attribute describes the quality focus of the proposed method.

Attribute values:

- Security: The focus is on security.
- Safety: The focus is on safety.
- General: The method is not specifically geared to security, but it has at least a partial focus on security.

Attribute: **Domain**

This attribute characterizes the domain in which the proposed method can be applied.

Attribute values:

- Software: The indicator specifically addresses software, software configuration, or software engineering processes.
- System: The indicator is not confined to software issues, but can be applied to other system or process aspects as well. The system may be any combination of software, hardware and network.

Attribute: **Context**

This attribute describes whether the proposed indicators are generic or context specific.

Attribute values:

- Context specific: The proposed indicator(s) can only be applied to one specific context or technology, e.g. to an OSGi platform or to peer-to-peer applications, etc.
- Generic: The approach is generic and not limited to a specific context.

Attribute: **Application Scope**

This attribute describes whether the proposed indicators refer to static or dynamic properties of the target.

Attribute values:

- Static: The indicator(s) can be applied to structural, immutable properties of the target, such as source code, system architecture, or process structure. The artifacts can be checked without executing the target (a system or a process). The indicator value does not change in operation, but is reproducible.
- Dynamic: The indicator refers to a property that dynamically changes during operation of the evaluation target — either because system attributes change or because the evaluation criteria are constantly evolving (e.g., the list of required patches). It must be applied at execution time of a system or process to assess current performance.

Attribute: **Purpose**

This attribute describes the intended use of the proposed indicator(s). More than one value may be chosen.

Attribute values:

- Measure: The target's security can be quantified and measured using adequate metrics, a ranking system or maturity levels.
- Assess: The target's security can be measured and compared to evaluated criteria to check the fulfillment of the criteria.
- Monitor: The target's security can be regularly reevaluated to identify irregularities or deviations from an established baseline.
- Examine: The target's security can be regularly assessed.
- Specify: The target's security properties can be described more objectively.
- Improve: The target's security can be enhanced systematically. The indicator reveals potentials for improvements and guides their implementation.
- Estimate: The target's security is not measured quantitatively but only qualitatively based on qualitative evaluation criteria.

A differentiation of the first four attribute values is shown in Figure 1.

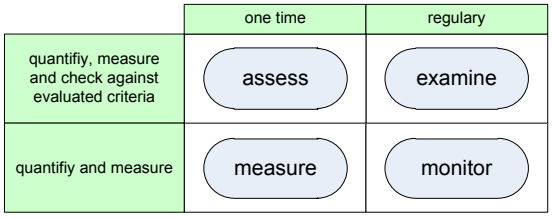

Figure 1 Differentiation of values for the purpose attribute

Attribute: **Dissemination**

This attribute characterizes the maturity of the proposed indicator(s).

Attribute values:

- Scientific: The proposed method originates from a study or research project and, to the best of our knowledge, has not been broadly applied in practice, yet.
- Applied: The proposed method is prevalent, approved and has been applied in practice.

- (Quasi-) Standard: The approach was accepted or developed by a standardization committee. If it is considered a quasi-standard by a broad audience, the »Standard« tag can also be chosen.

Attribute: **Tool Support**

This attribute categorizes available tool support. More than one value may be chosen.

Attribute values:

- None: To the best of our knowledge, no tool support is available.
- Academic: A prototype tool has been developed that supports the proposed indicator(s).
- Commercial: A commercial tool exists that supports the indicator(s).
- Free For Use: A freeware tool is available that supports the indicator(s).
- Open Source: A tool was developed in an open source project.

Attribute: **Life Cycle Phase**

This attribute describes which phase or phases in the life cycle of the target the indicator affects. More than one value may be chosen. If no explicit point of appliance is defined in the classified method's publication, the most appropriate phases are selected. If the method can be applied independently from any life cycle restrictions, all phases are selected.

Attribute values:

- Requirement
- Design
- Implementation
- Testing
- Deployment
- Operation

Attribute: **Functionality Classes**

This attribute defines functionality classes to which the proposed indicators can belong. Each work addresses at least one functionality class. Functionality classes cover all relevant ways in which an indicator can affect the security of a system. By categorizing the approaches into these functionality classes, trends or prevalent combinations of functionality classes can be revealed.

- Security Requirement Definition: The approach guides the specification of adequate security requirements, pinpoints concrete security requirements, or controls a process to define adequate security requirements. The pro-

posed indicator(s) can indicate the quality of the elicited security requirements or of the requirements elicitation process.

- Identification: The approach guides the identification of security issues (e.g. threats, vulnerabilities, security defects in source code, shortcomings in processes relevant for security, etc.).
- Measurement: The proposed indicator(s) can be used to quantitatively or qualitatively measure, assess, monitor or examine security relevant aspects. The approach provides at least one security metric or security measurement, or detailed guidance to develop context specific security metrics or security measurements to quantify security aspects.
- Mitigation: The approach suggests concrete treatments to mitigate a specific security defect that can be located in a design or management process, a system design, or a running system. Furthermore, there is a clear causal relation or correlation between defect and suggested countermeasure (in contrast to mere improvement approaches that suggest general improvement potentials — e.g., best practices — but do not address a specific vulnerability).
- Best Practice: The approach provides a catalog of best practices or security principles related to security. The focus of the proposed indicator(s) is on the way of improving security, the means to an end. In our classification security process definitions or improvements belong to the Best Practice class, too.
- Criteria Catalog: The approach provides a criteria catalog that contains security related criteria, which can be used for security requirements definition or security validation. The focus is on desirable security properties, that is, on security goals (as opposed to best practices to achieve these goals).
- Return on Investment Calculation: The approach introduces a methodology to calculate the return on investment in security.

3.2 Classification

This section contains an overview of the body of work that has been classified according to our classification model. The various indicators proposed in these publications will be described in more detail in our internal report [RS12]. Table 1 lists the abbreviations used in the classification overview. Table 2 presents the results of the classification. The approaches are ordered alphabetically by name. The attributes were filled out to the best of our knowledge.

Table 1: Abbreviations in the Classification Overview

Target	
Process	Pc
Product	Pd
Resource	Re

Application Scope	
Static	S
Dynamic	D

Focus	
Security	Sec
Safety	Saf
General	G

Dissemination	
Scientific	Sc
Applied	Ap
(Quasi-) Standard	St

Domain	
System	Sys
Software	SW

Tool Support	
None	N
Academic	A
Free For Use	F

Context	
Context Specific	Sp
Generic	Gc

Table 2: Classification of the surveyed approaches

Name	Reference	Target	Focus	Domain	Context	Application Scope	Dissemination	Tool Support	Purpose							Usage in the Life Cycle						Functionality Classes						
									Measure	Assess	Monitor	Examine	Specify	Estimate	Improve	Requirements	Design	Implementation	Test	Deployment	Operation	Security Requirement	Identification	Measurement	Mitigation	Best Practice	Criteria Catalog	Return on Investment
A Guide to Building Secure Web Applications and Web Services	OWA05	Pd	Sec	Sys	Sp	S	Ap	N					X		X	X	X	X					X		X	X		
A Metrics Framework to Drive Application Security Improvement	NP07	Pd	Sec	Sys	Sp	D, S	Sc	N	X								X			X	X			X				
A Model of Return on Investment for Information Systems Security	AD03	Pc	Sec	Sys	Gc	D	Sc	N	X							X												X
Adaptive Vulnerability Analysis	VGM+96	Pd	Sec	SW	Gc	D	Sc	A	X										X					X				
An Attack Surface Metric	MW05 MTM+07	Pd	Sec	Sys	Gc	S	Sc	N	X						X				X				X	X	X			
CCD	SMW+11	Pd	Sec	SW	Gc	S	Sc	A	X		X							X	X				X	X				
CISWG Report of the Best Practices and Metrics Team	CISWG04	Pd	Sec	Sys	Gc	D	St	N			X		X			X					X	X		X		X		
COBIT	ITGI07	Pc, Re	Saf, Sec	Sys	Gc	S	Ap	N				X			X		X	X		X	X					X		
Common Criteria	CC09 CC09a CC09b	Pd	Sec	Sys	Gc	S	St	N					X	X		X			X			X						X	

Name	Reference	Target	Focus	Domain	Context	Application Scope	Dissemination	Tool Support	Purpose							Usage in the Life Cycle						Functionality Classes						
									Measure	Assess	Monitor	Examine	Specify	Estimate	Improve	Requirements	Design	Implementation	Test	Deployment	Operation	Security Requirement	Identification	Measurement	Mitigation	Best Practice	Criteria Catalog	Return on Investment
Complete Guide to Security and Privacy Metrics	Her07 Sow10	Pd	Sec	Sys	Gc	D, S	Ap	N	X							X	X	X	X	X	X			X				X
CVSS	MSR07 WXZ07 PPN06	Pd	Sec	Sys	Gc	S	Ap	F	X							X	X	X	X	X	X			X				
CWSS	Mit11	Pd	Sec	Sys	Gc	S	Sc	N	X							X	X	X	X	X	X			X				
DITSCAP	ASD00	Pc, Pd	Sec	Sys	Gc	S	St	N					X	X		X	X		X		X	X					X	
eSAFE	GoI10 GoI10a	Pc, Pd	Sec	Sys	Gc	S	St	N	X				X	X		X	X	X			X	X		X			X	
Estimating a System's Mean Time-to-Compromise	LB08	Pd	Sec	Sys	Gc	D	Sc	N	X												X			X				
Framework for Measuring and Reporting Performance of Information Security Programs in Offshore Outsourcing	Set06	Pc, Pd	Sec	Sys	Gc	D	Sc	N				X									X			X				
GAISP	ISSA04	Pd	Sec	Sys	Gc	S	St	N					X	X		X										X		
IT-Grundschutz	BSI05 BSI09	Pc, Pd	Sec	Sys	Gc	S	St	C					X	X	X	X	X	X			X	X			X	X	X	
ITIL	Bru06 Olb04 itSMF07	Pc	G	Sys	Gc	S	St	N					X		X		X				X		X			X		

Name	Reference	Target	Focus	Domain	Context	Application Scope	Dissemination	Tool Support	Purpose							Usage in the Life Cycle						Functionality Classes						
									Measure	Assess	Monitor	Examine	Specify	Estimate	Improve	Requirements	Design	Implementation	Test	Deployment	Operation	Security Requirement	Identification	Measurement	Mitigation	Best Practice	Criteria Catalog	Return on Investment
MASTER	LPF+08	Pd	Sec	SW	Gc	D	Sc	N			X		X			X	X	X			X			X				
Microsoft Security Development Lifecycle	Mic10 Mic10a Mic05	Pc	Sec	SW	Gc	S	Ap	N					X	X		X	X	X	X	X						X		
NIST 800-100	BHW06	Pc	Sec	Sys	Gc	D, S	St	N					X		X	X	X	X	X	X	X					X		
NIST 800-27	SHF04	Pd	Sec	Sys	Gc	S	St	N					X	X		X	X	X	X	X	X	X				X		
NIST 800-30	SGF02	Pc, Pd	Sec	Sys	Gc	S	St	N	X				X		X	X	X	X			X		X	X		X		
NIST 800-53	RSP+09	Pc, Pd	Sec	Sys	Gc	S	St	N					X	X		X						X					X	
NIST 800-53A	RSS+10	Pc, Pd	Sec	Sys	Gc	D, S	St	N					X	X		X			X	X		X		X			X	
NIST 800-55	CSS+08	Pc, Pd	Sec	Sys	Gc	D, S	St	N			X		X			X	X	X	X	X	X			X		X		
OSSTMM	Her10	Pc, Pd	Sec	Sys	Gc	D, S	St	N	X				X						X					X				
OWASP ASVS	OWA09	Pd	Sec	Sys	Sp	S	St	N		X			X	X		X	X		X			X		X			X	

Name	Reference	Target	Focus	Domain	Context	Application Scope	Dissemination	Tool Support	Purpose							Usage in the Life Cycle						Functionality Classes						
									Measure	Assess	Monitor	Examine	Specify	Estimate	Improve	Requirements	Design	Implementation	Test	Deployment	Operation	Security Requirement	Identification	Measurement	Mitigation	Best Practice	Criteria Catalog	Return on Investment
OWASP Testing Guide v3	MKC+08	Pd	Sec	Sys	Sp	D, S	Ap	N						X		X	X	X	X	X	X					X		
PCI DSS v1.2.1	PCI10	Pc, Pd	Sec	Sys	Sp	S	St	N					X	X		X				X	X	X					X	
Performance Metrics for Information Security Risk Management	RR08	Pd	Sec	Sys	Gc	D	Sc	N	X							X												X
Practical Measurement Framework for Software Assurance and Information Security	BMB+08	Pc, Pd	Sec	Sys	Gc	D, S	Sc	N	X							X	X	X	X		X			X				
Risk-based Security Engineering through the Eyes of the Adversary	EW05	Pd	Sec	Sys	Gc	S	Sc	N	X							X	X	X	X	X	X		X	X				
Security benchmarks of OSGi plattforms	PF08	Pd	Saf, Sec	SW	Sp	S	Sc	N	X					X	X				X					X			X	
Security Metric Architecture for Next Generation Network	HYY09	Pd	Saf, Sec	Sys	Gc	D	Sc	C, F	X												X			X				
Security Patterns	SNL06 DSS+09	Pd	Sec	SW	Gc	S	Ap	N					X				X									X		
SGIT	PJM08	Pd	Sec	Sys	Gc	S	Sc	A						X		X	X	X	X				X			X	X	
SSE-CMM	CMU03	Pc	Sec	Sys	Gc	S	Ap	N		X			X		X	X	X	X	X	X	X					X		

Name	Reference	Target	Focus	Domain	Context	Application Scope	Dissemination	Tool Support	Purpose							Usage in the Life Cycle						Functionality Classes						
									Measure	Assess	Monitor	Examine	Specify	Estimate	Improve	Requirements	Design	Implementation	Test	Deployment	Operation	Security Requirement	Identification	Measurement	Mitigation	Best Practice	Criteria Catalog	Return on Investment
Static Code Analysis	CW07 DMS07 HL03 WNZ+07	Pd	Sec	SW	Gc	S	Ap	C, O						X	X				X				X		X	X	X	
Threat Modeling for CSRF Attacks	LZR+09	Pd	Sec	SW	Sp	S	Sc	N					X		X		X						X		X			
Total Return on Investment	Pur04	Pc	Sec	Sys	Gc	D	Sc	N			X										X							X
Trust4All	T4A06	Pd	G	Sys	Gc	D, S	Sc	N	X										X		X			X				

3.3 Remarks

Security is rarely included in general quality assurance approaches: Typically, it is only treated in passing or »left as an exercise for the reader«. This leads to the question whether quality assurance experts see security as an easy to solve problem for which the general quality assurance approaches are adaptable in an obvious way or whether they lack experience in the security domain and just assume that their approaches should be applicable there, too. From our point of view, both statements are disputable as we believe that security differs from other quality aspects in many respects. Therefore, it requires specific quality assurance approaches that differ from those for other system qualities. Our claim is supported by the fact that even safety and security — two qualities that have much in common at first glance — are mostly treated independently, and only few works offer an integrated approach.

Most indicators proposed in standards are either purely qualitative, or they are restricted to the following, quite simplistic format:

> **{ Count the number of | Determine the percentage of } systems**
> **{ where <some security measure> has been applied | that satisfy <some simple criterion from a best practice catalog> }.**

Hardly any software-(configuration-)specific indicators exist: Most approaches address processes or overall system, but few provide applicable security metrics for secure software engineering, i.e., static indicators that could predict the security »quality in use« of system architectures or source code in advance of deployment.

Table 3 summarizes the result of our grouping into functionality classes. The figures may be deceptive as they are strongly related to our (somewhat subjective) selection of representative works in the field of »security indicators«.

Approximately half of the analyzed approaches (20 out of the 44) provide some kind of »measurement« methodology. 16 works can be seen as best practice guides that aim at enhancing the security of systems, but only three publications propose both, »security measurement« and »best practice guidance«. This reveals a lack of verification that the proposed best practice guidelines really benefit security.

Besides measuring, eight works propose indicators that aim at identifying concrete security issues. Together, this makes 25 works that help to uncover possible security flaws (three publications have both interests). Contrary to this, only five approaches give concrete suggestions to mitigate the risks caused by these security issues. In our opinion, the identification should go hand in hand with proposals for remediation or at least best practices that result in measureable security improvements.

Only six methods we surveyed have tool support. Tools are only available for methods with a generic context and a focus on the target »product«.

Many threat and vulnerability modeling techniques have been developed. Most of them are purely descriptive. They cannot be seen as security indicators as they do not measure or quantify security at all, but are only meant to create a graphical representation of security issues. Such approaches have been excluded from this survey as they do not directly support measurability.

Table 3 Assignment of indicator approaches to functionality classes

Class	Number of approaches belonging to that class
Security Requirement Definition	10
Identification	8
Measurement	20
Mitigation	5
Best Practice	16
Criteria Catalog	11
Return on Investment Calculation	4

4 Classification of Security Indicators

From the information gathered during the literature review and the former classification attempt, we designed a classification tree for security indicators. All approaches that we identified in our survey can be placed in the tree as nodes according to their purpose of measurement. The indicators can then be added as leaves to their corresponding approaches. As indicators could cover more than one purpose or appear in different approaches, they may occur redundantly in the tree.

The classification tree discloses dependencies between the analyzed approaches. It reveals the grouping and distribution of research and standardization efforts over the last decade.

Furthermore, the classification tree can help to select a proper security indicator for a given scenario. In combination with the classification model for the surveyed approaches, the most promising candidates out of a group of available approaches for indicator-based security control should be identifiable.

For clarity, not all individual indicators found in the literature, but only typical representatives per indicator class are placed in the classification tree. Our entire tree is shown in our internal report [RS12].

4.1 Description of the Classification Tree

The classification tree is divided into five main branches representing the five major classification categories. Each category represents a different aspect of security measurement. Figure 2 shows the main branches of the classification tree. Below, we describe these branches in more detail and give examples that illustrate each category. The little circular symbol at the end of each branch symbolizes an expandable sub-tree.

Figure 2: Main categories of the classification tree

The following symbols are used in the tree:

- **Pencil:** Symbolizes a measurable attribute.
- **Magnifier:** Reference to work that has been analyzed in this survey.
- **Star:** Symbolizes a concrete security indicator (connected to the source where this indicator is described).
- **Magic Wand:** Symbolizes a security indicator that is not explicitly mentioned in the publication but suggests itself in the given context.
- **Red Arrow:** Marks a reference to a redundant branch in the tree.

4.1.1 Compliance

The Compliance branch lists all security indicators that measure the degree to which security requirements are met by a security target (system or process). Some methods derivate a maturity index as a compliance measure. The main branch is depicted in Figure 3.

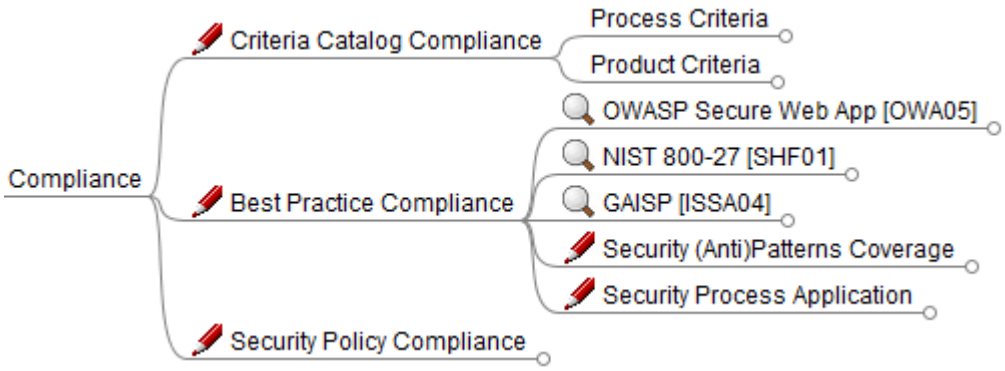

Figure 3: Compliance branch

The Compliance branch is divided into three sub-trees. The Criteria Catalog Compliance sub-tree contains all security indicators that measure the compliance of a single security target with a set of required security characteristics listed in a Criteria Catalog. The typical format of these indicators is:

> **{ Percentage | Number } of required security criteria satisfied by an individual <security target>**

As an example, the tree for Criteria Catalog Compliance with the relevant literature references is shown in Figure 4. The other two sub-trees contain measures that determine the compliance rate to best practice catalogs and security policies. The typical format is similar to the Criteria Catalog Compliance.

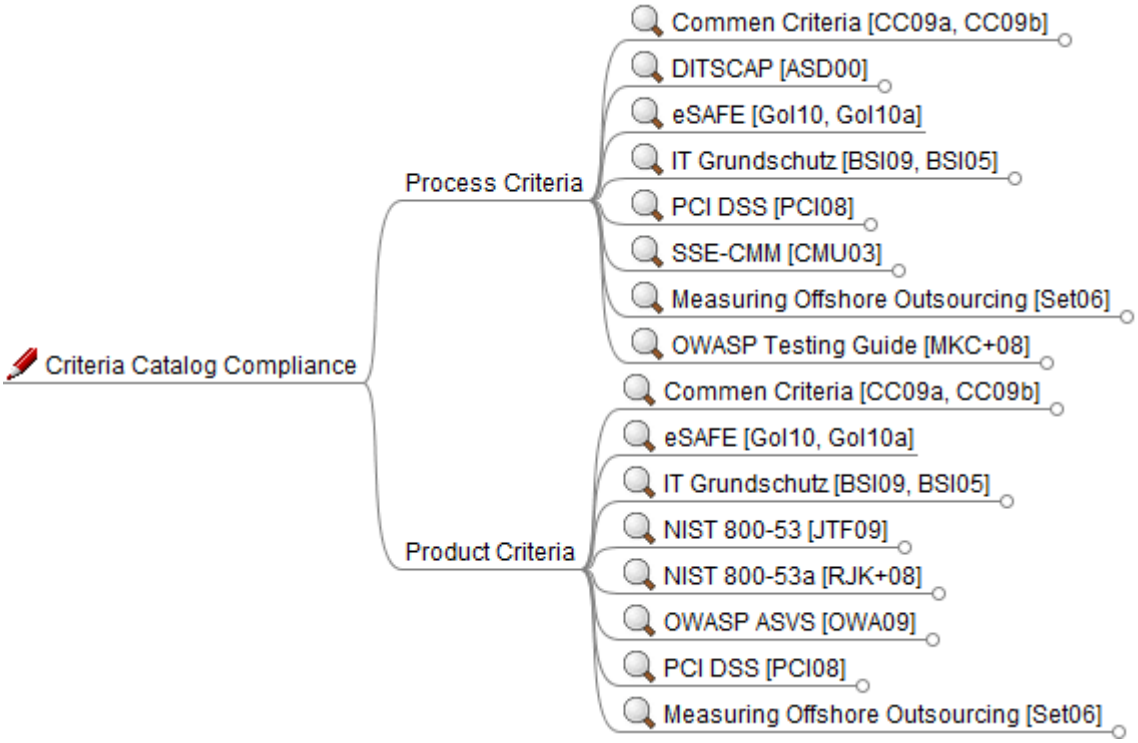

Figure 4: Compliance branch – Criteria Catalog Compliance

Figure 5 shows exemplary measurements to be applied on products. Most of them are not directly presented in the publications, but can be derived from the context. As this branch only lists criteria catalogs for compliance checking, the proposed measurements describe the fraction of implemented or assessed security features. In addition, some measures rate the completeness of the implementation of proposed security features on a predefined security level scale.

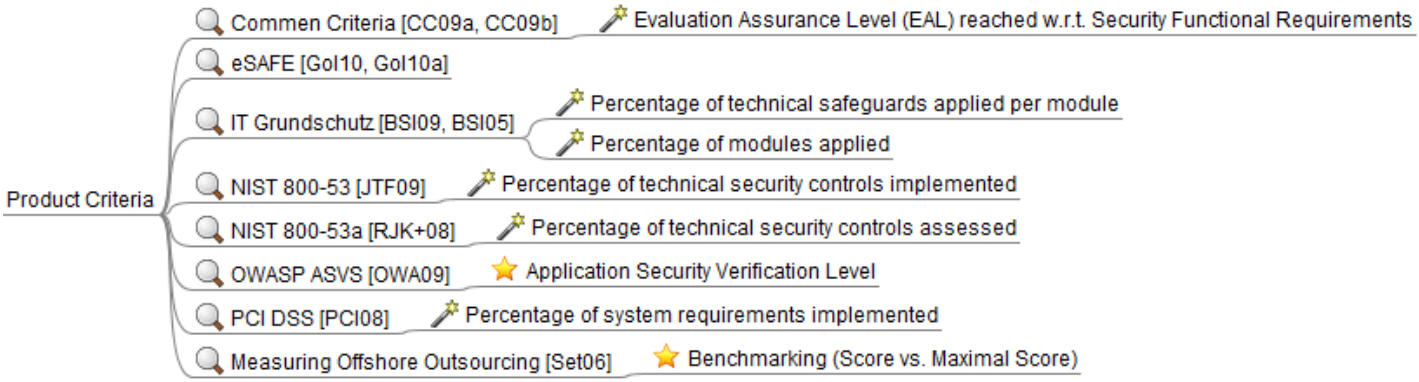

Figure 5: Compliance branch – Product Feature Indicators

4.1.2 Target Coverage

The Target Coverage branch comprises all security indicators that measure the fraction (or the absolute number) of security targets that satisfy a given security criterion. The branch is divided into three sub-trees that contain measures for patch, policy and protection coverage. The typical format is:

{ Percentage | Number } of security targets satisfying <security criterion>

The main branch is depicted in Figure 6.

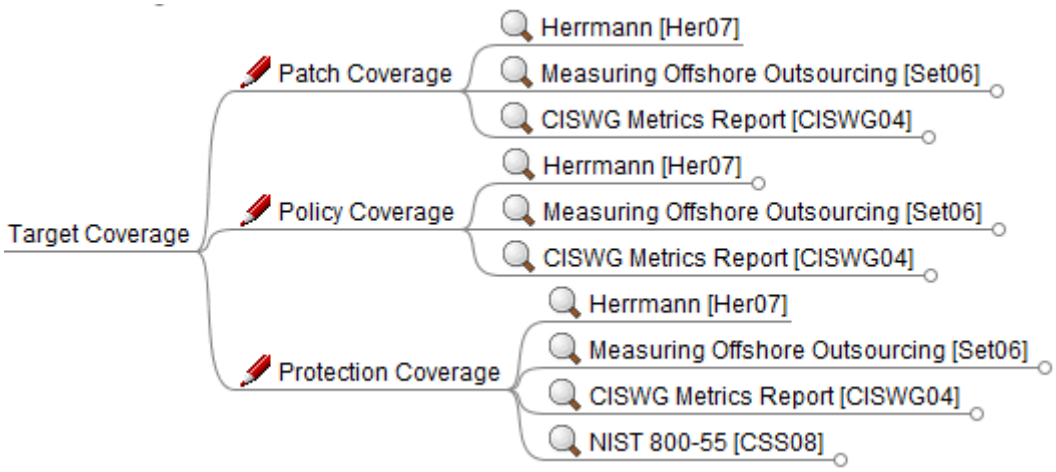

Figure 6: Target Coverage branch

Note that Compliance and Target Coverage complement each other. While the former measure the coverage of a given set of security requirements, the latter measures the coverage of the set of target systems: Ideally, every applicable requirement should be applied to every affected target system.

As an example, the Policy Coverage sub-tree is shown in Figure 7.

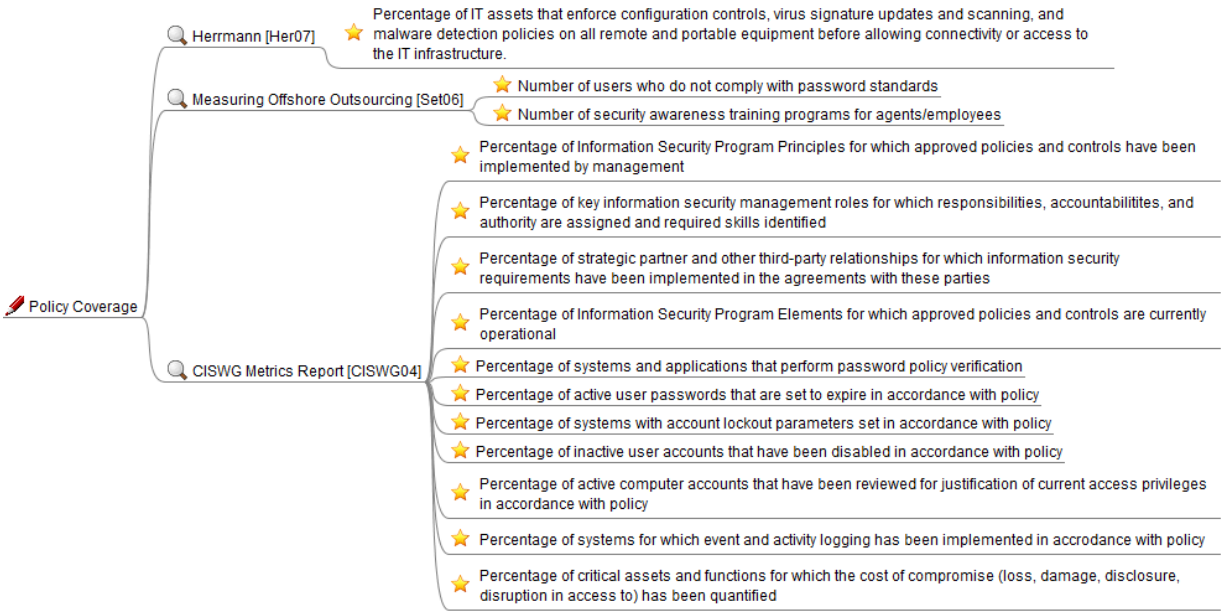

Figure 7: Coverage branch – Policy Coverage

4.1.3 Cost

The Cost branch is split in several matters of expense, which are the costs of an attack for the adversary (Attack Cost), the costs to prevent attacks (Countermeasure Cost), and the sustained losses and damage caused by security incidents including the costs for their remediation. Security indicators that measure or estimate such costs can be placed in the tree structure shown in Figure 8.

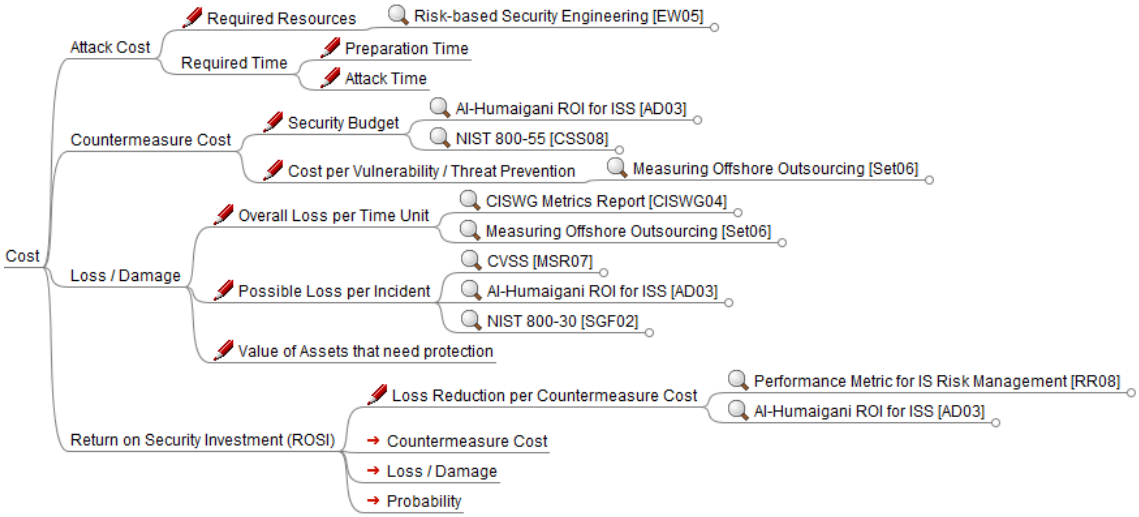

Figure 8: Cost branch

Figure 9 shows several measurements that can support the assessment of costs caused by security incidents and vulnerabilities.

Figure 9: Cost branch – Possible Loss per Incident indicators

4.1.4 Probability

The Probability branch, which is depicted in Figure 10, contains security indicators that measure or estimate the likelihood of attack attempts. Most indicators reflect factors that attract or repel the adversary, or that alleviate or aggravate attacking. The »attack frequency« indicator tries to extrapolate from the observed number of incidents in the past to the predicted number of similar incidents in the future.

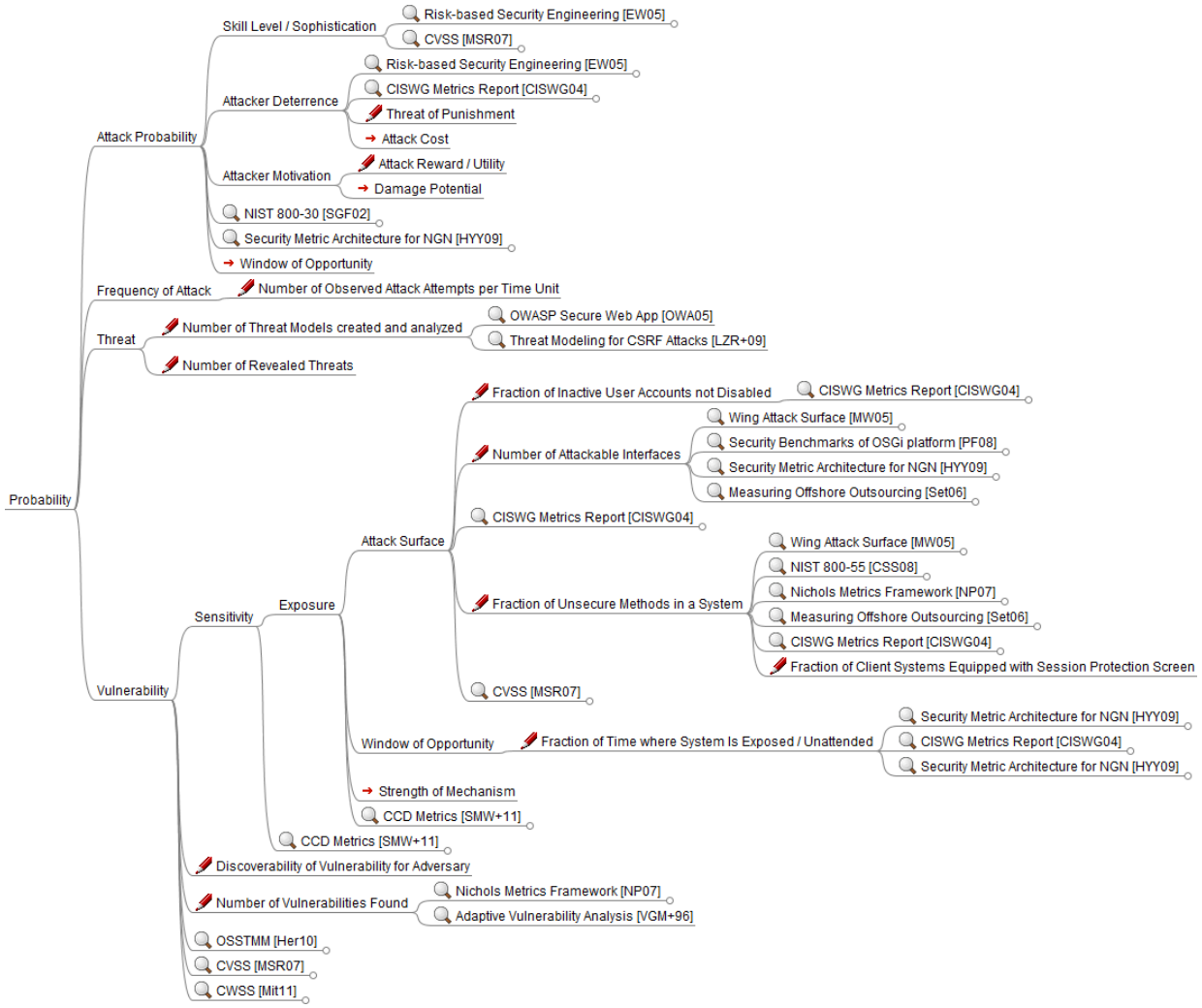

Figure 10: Probability branch

As an example for measuring the probability of vulnerability exploitation, exemplary measures from the Attack Surface Branch are shown in Figure 11.

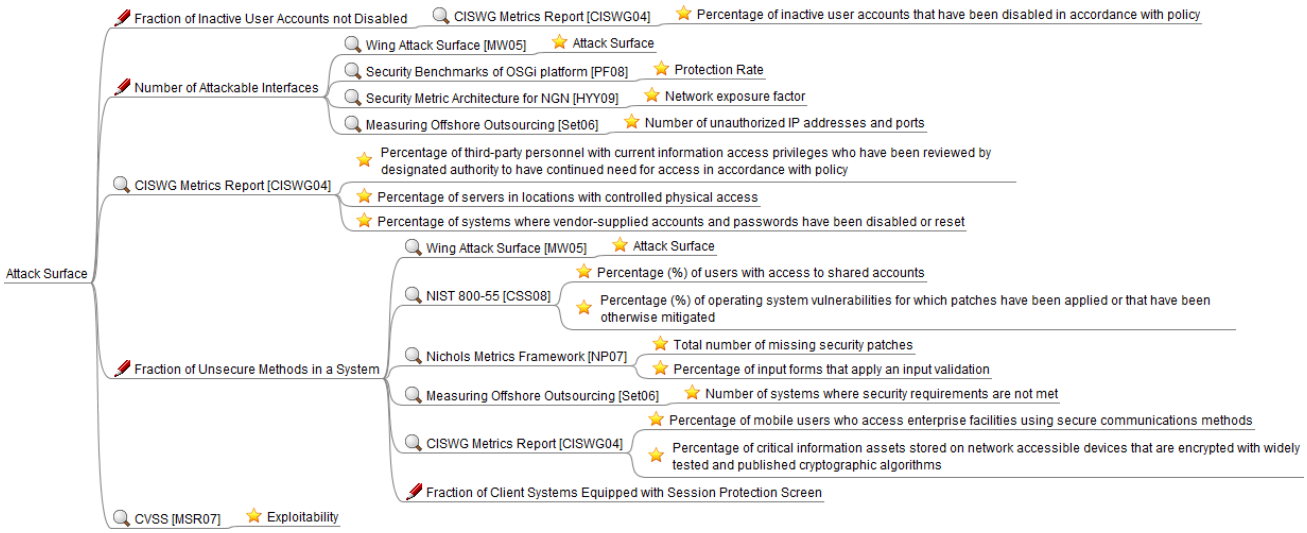

Figure 11: Probability branch — Attack Surface indicators

4.1.5 Effectiveness / Rigor

The Effectiveness and Rigor branch contains all security indicators that measure or estimate the success rate of countermeasures against attacks, as shown in Figure 12. This branch is divided into three countermeasure categories:

- Protection: security indicators that determine the effectiveness of attack avoidance
- Detection: security indicators that measure the success rate and delay of detection mechanisms to reveal successful attacks or attack attempts
- Response: security indicators that measure the effectiveness and speed of incident response

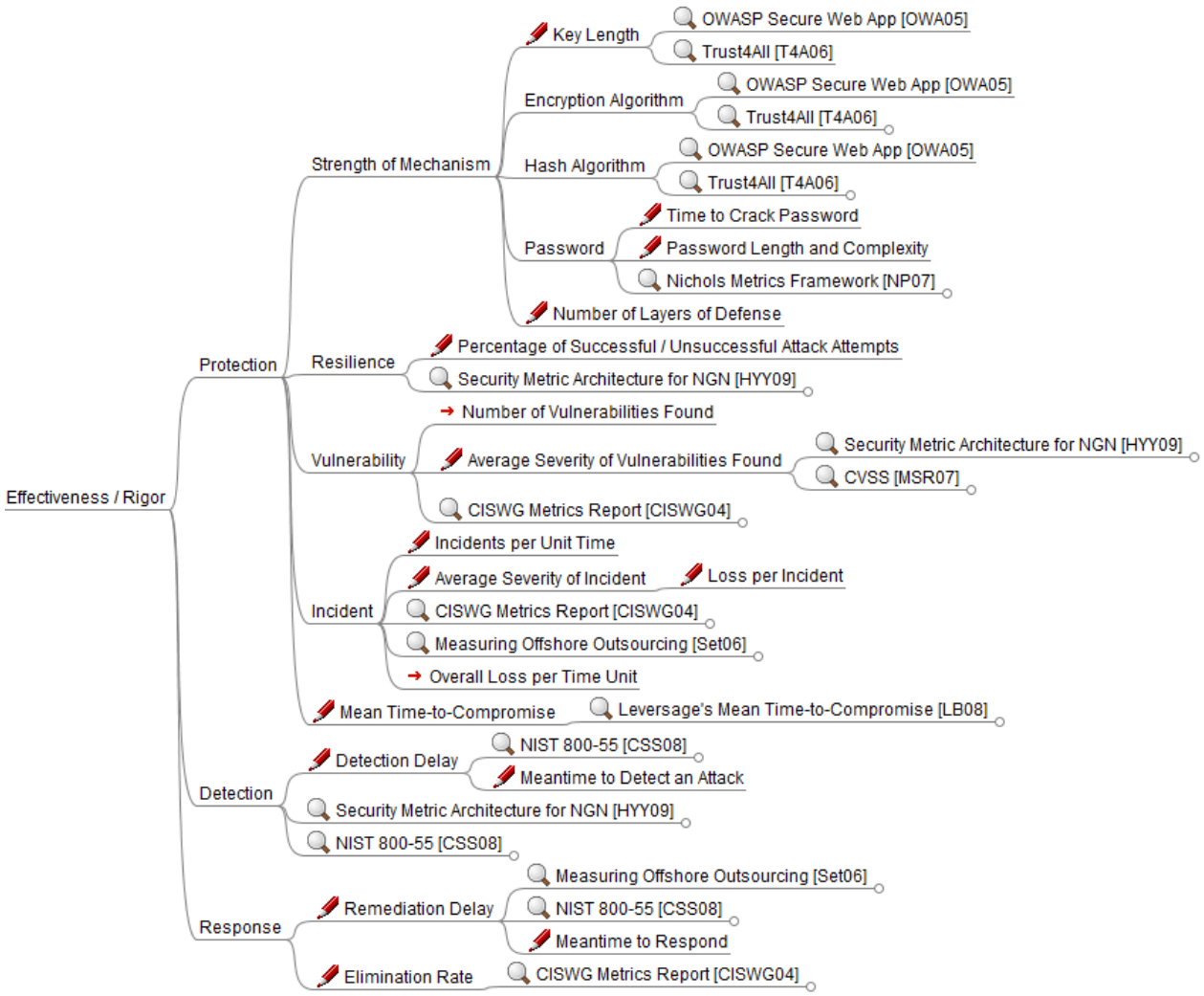

Figure 12: Effectiveness and Rigor branch

A measure for protection effectiveness is, for example, the »Strength of Mechanism«, which indicates the security of applied encryption and hash algorithms, key length and password quality etc. As effective security indicators, the key length of these mechanisms can be compared to federal recommendations; the time to by-pass a security mechanism can be estimated. Strength measures should be used during the development of a system; they require periodic re-evaluation after deployment to account for a continual increase in attack capabilities, for example, due to growing computation speed or progress in cryptology.

In operation, indicators such as »Vulnerability Discovery Rate« or »Meantime to Compromise« reveal the effectiveness of the implemented security mechanisms. They may also serve as stop criteria for penetration testing.

Exemplary measurements to rate the frequency and severity of security incidents — an indicator for the effectiveness of protective countermeasures — are presented in Figure 13.

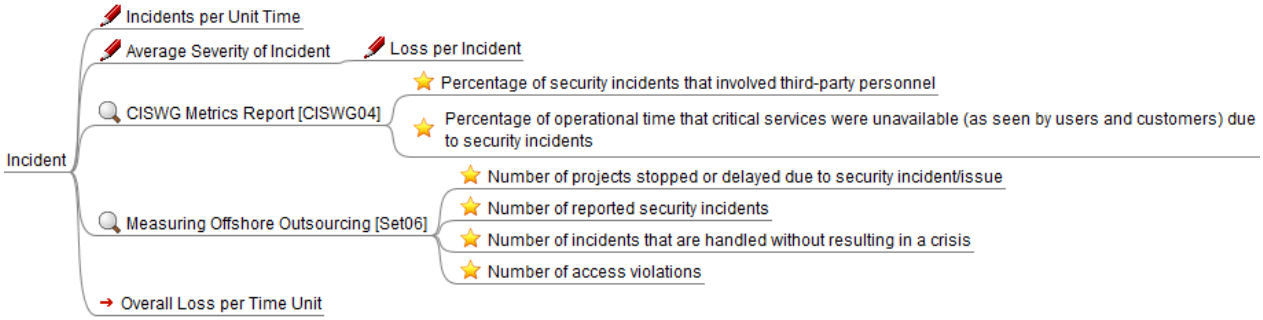

Figure 13: Effectiveness and Rigor branch – Incident Indicators

4.2 Indicator Characterization

Based on our classification tree of available security indicators, we finally extracted the main characteristics of the proposed indicator types from the surveyed works and characterized the most representative indicators according to these properties. In Section 4.2.1, we first describe the attributes that we chose for our characterization; Section 4.2.2 presents the table of indicators, classified according to these attributes.

4.2.1 Attribute Description

This section defines the attributes that we used for the categorization of the indicators, and it describes the attribute values they can take.

Attribute: **Name**

This attribute specifies the name of the indicator or a descriptive phrase if an established name is not provided in the literature.

Attribute: **Target**

This attribute denotes the target on which the indicator can be applied. More than one value may be chosen.

Attribute values:

- Product: The indicator refers to the security of an IT system, a piece of software (source code, components), or a network.
- Process: The indicator refers to security-related properties of a process, e.g., a development process or a maintenance process. The underlying rationale is that improving the process will indirectly improve the security of the product.

- Resources: The indicator refers to security-related attributes of resources that are used during development, deployment, or operation of a product. Resources can be, for example, supporting tools, personnel, or material used in production.

Attribute: **Target Changeability**

This attribute describes the extent to which the indicator is affected by changes of the evaluation target. A modification in the target can influence the results of a measurement and therefore affects the required measurement frequency.

Attribute values:

- No Changes: The target's attributes relevant for the indicator can be assumed to remain unchanged. For example, the *key length of a smartcard's cryptochip* typically won't change after an initial design decision has been taken.
- Static Changes: Structural properties of the target relevant for the indicator can change. For example, the *adequacy of the current patch level* depends on the number and type of patches installed.
- Dynamic Changes: The runtime behavior of a target changes properties that are relevant for the indicator. For example, the *number of critical log entries* is a transitional security attribute that constantly changes during operation.

Attribute: **Measure Changeability**

This attribute describes the changeability of the measure, respectively, the interpretation of the measurement result.

Attribute values:

- No Changes: The measure and its interpretation do not change. For example, the *percentage of process criteria covered* is a fixed property of a given process.
- Predictable Changes: Changes in the interpretation of the measurement results are (assumed to be) predictable as they follow unwritten or natural laws. For example, for a symmetric encryption key the *adequate key length* can be assumed to increase by roughly one bit per year (if we assume that according to Moore's Law the computation power for a brute force attack doubles every year).
- Unpredictable Changes: Changes in the interpretation of the measurement results are unpredictable as they are caused by unforeseen incidents. For example, the *adequacy of virus signatures* depends on the occurrence of new virus attacks or the identification of new vulnerabilities.

Attribute: **Measurement Frequency**

This attribute describes the recommended measurement frequency according to the target changeability and the measure changeability.

Attribute values:

- Once: It typically suffices to measure only once because it can be assumed that the measurement result is not affected by measure changeability.
- Event-Triggered: The measurement is triggered by an irregularly occurring event that changes the measure, such as a new breakthrough in cryptology
- Time-Triggered: The indicator must be applied periodically as it may be affected by dynamic target changes or unpredictable measure changeability.

Attribute: **Measure / Assess**

This attribute describes the intended use of the indicator.

Attribute values:

- Measure: The target's security can be quantified and measured using adequate metrics, a ranking system or maturity levels.
- Assess: The target's security can be measured and the measurement result can be compared to evaluated criteria to check the fulfillment rate of the criteria.

Attribute: **Descriptive / Constructive**

This attribute describes the extent to which the indicator guides the user in taking adequate security actions.

Attribute values:

- Descriptive: The indicator describes the status of a security-related property, but it does not indicate countermeasures that could help to improve the current security status. For example, the *number of security incidents within observation period* describes the (lack of) security of a system, but does not help in the identification and removal of the vulnerabilities that were exploited.
- Constructive: The measurement result suggests adequate steps that would help to improve the current security status. For example, the *percentage of systems having the latest security patches installed* suggests a concrete improvement: Patch the remaining systems that have an inadequate patch level!

Attribute: **Life Cycle Phase**

This attribute describes in which phase or phases in the life cycle of the target the metric can be applied. More than one value may be selected. If no explicit point of appliance is defined in the classified indicator's publication, the most appropriate phases are assigned. If the method can be applied independently from any life cycle restrictions, all phases are selected.

Attribute values:

- Requirement
- Design
- Implementation
- Test and Inspection
- Deployment
- Operation

Attribute: **Measurement Goal**

This attribute defines functionality classes to which the indicator can belong. These classes cover the relevant ways in which an indicator can affect the security of a system.

- Security Requirements Definition: The measurement result helps to guide the specification of adequate security requirements, to pinpoint concrete security requirements, to control the requirements elicitation process, or to indicate the quality of the security requirements.
- Identification: The measurement result helps to identify security issues and their causes (e.g. threats, vulnerabilities, security defects in source code, shortcomings in processes relevant for security, etc.).
- Mitigation: The measurement result helps to control attributes of a product, or process so that security risks during development or operation are prevented. That is, the measurement not only identifies potential shortcomings, but it also indicates adequate countermeasures.
- Compliance / Coverage: The indicator determines the degree of compliance to a catalog of best practices or security criteria, or the degree to which a the implementation of a given security requirement covers the set of affected target systems.
- Return on Investment Calculation: The indicator measures the return on investment in security or the measurement result can be used as one parameter in a return on investment calculation.

4.2.2 Indicator Categorization

Below, we characterize each indicator with respect to the attributes described in Section 4.2.1. For our summary table we selected the most representative types of indicators for each category from our classification tree (cf. Section 4.1 and our internal report [RS12]).

Table 5 summarizes our classification result. Abbreviations used in Table 5 can be found in Table 4. The aim of the categorization was to determine the coverage of various security aspects, life-cycle phases and usage scenarios by the known security indicators.

Table 4: Abbreviations used in the indicator categorization (Table 5)

Target	
Process	Pc
Product	Pd
Resource	Re
Measure Changeability	
No Changes	N
Predictable Changes	P
Unpredictable Changes	U
Measure / Assess	
Measure	M
Assess	A

Target Changeability	
No Changes	N
Static Changes	S
Dynamic Changes	D
Measurement Frequency	
Once	O
Event-Triggered	ET
Time-Triggered	TT
Descriptive / Constructive	
Descriptive	D
Constructive	C

Table 5: Indicator classification

Name	Target	Target Changeability	Measure Changeability	Measurement Frequency	Measure / Assess	Descriptive / Constructive	Usage in the Life Cycle						Measurement Goal				
							Requirements	Design	Implementation	Test and Inspection	Deployment	Operation	Security Requirement	Identification	Mitigation	Compliance / Coverage	Return on Investment
Compliance: Criteria Catalog, Best Practice, Security Policy																	
Percentage of Process Criteria covered (e.g., CC Assurance Requirements)	Pc	S	N	ET	A	C	X	X	X	X	X	X	X	X	X	X	
Percentage of Product Security Criteria covered (e.g., CC SFR, GSHB Maßnahmen M1, M4, M5)	Pd	S	N	ET	A	C	X	X	X	X	X		X	X	X	X	
Percentage of Security Artifact Criteria covered (e.g., SSE-CMM, quality of contingency plan)	Re	S	N	ET	A	C	X	X	X	X	X	X	X	X	X	X	
Process or Product Maturity Level (e.g., SSE-CMM Maturity Level, Common Criteria EAL Level)	Pd, Pc	S	N	ET	A	C	X	X	X	X	X	X	X	X	X	X	
Product Security Benchmark Score (e.g., Router Audit Tool Benchmark)	Pd	D	N	TT	A	C			X	X	X	X		X	X	X	
Percentage of Security Best Practices followed (e.g., SSE-CMM Practices, Microsoft SDL)	Pc	S	N	ET	A	C	X	X	X	X	X	X	X	X	X	X	
Percentage of Security Policy complied to	Pc, Pd	D	N	TT	A	C					X	X		X	X	X	
Percentage of Policy Audits without violations noted	Pc, Pd	D	U	TT	M	C					X	X	X	X	X	X	
Percentage of applicable regulations covered during policy audit	Pc, Pd	D	U	TT	M	C					X	X	X	X	X	X	
Target Coverage: Patch Coverage, Policy Coverage, Protection Coverage																	
Percentage of systems that comply to security policy (e.g., that satisfy all/specific criteria of the policy)	Pd	D	U	TT	A	C						X	X	X	X	X	

Name	Target	Target Changeability	Measure Changeability	Measurement Frequency	Measure / Assess	Descriptive / Constructive	Usage in the Life Cycle						Measurement Goal				
							Requirements	Design	Implementation	Test and Inspection	Deployment	Operation	Security Requirement	Identification	Mitigation	Compliance / Coverage	Return on Investment
Percentage of systems with safe-guards installed (e.g., virus scanner, patches)	Pd	D	U	TT	A	C						X		X	X	X	
Percentage of units with contingency planning	Pc, Re	D	N	TT	M	C						X		X	X	X	
Percentage of personnel with ade-quate/up-to-date security training	Re	D	N	TT	M	C	X	X	X	X	X	X		X	X	X	
Cost: Attack Cost, Countermeasure Budget, Loss and Damage																	
Cost of attack resources	Pd	N	N	O	M	D	X	X					X	X			
Time required for attack / mean time to compromise	Pd	N	N	O	M	D				X	X			X			
Security Budget available/spent (relative to required security budget per system or unit)	Pd, Re	D	N	TT	M	D				X	X	X					X
Percentage of budget devoted to information security	Re	D	N	TT	M	D				X	X	X					X
Overall loss per time unit due to security incidents (actual)	Pd, Pc, Re	D	N	TT	M	D						X		X			X
Average/Maximum loss per incident (actual)	Pd, Pc, Re	D	N	TT	M	D						X		X			X
Average loss per incident (expected: Likelihood times impact per threat)	Pd, Pc, Re	N	N	O	M	D	X	X					X	X			
Value of Assets that need protection	Pd	N	N	O	M	D	X							X			
Probability: Attack Probability, Frequency of Attacks, Threat Modeling																	
Skill level required for attack	Pd	N	N	O	M	D	X	X		X	X		X				X

Name	Target	Target Changeability	Measure Changeability	Measurement Frequency	Measure / Assess	Descriptive / Constructive	Usage in the Life Cycle						Measurement Goal				
							Requirements	Design	Implementation	Test and Inspection	Deployment	Operation	Security Requirement	Identification	Mitigation	Compliance / Coverage	Return on Investment
Deterrence (e.g., threat of punishment)	Pd, Pc, Re	N	N	O	M	D	X	X					X				
Attack reward / utility	Pd	N	N	O	M	D	X						X				
Size of attack surface	Pd	S	N	ET	A	C		X	X		X			X	X		
Time window of opportunity for attack	Pd, Pc, Re	D	N	TT	A	C		X			X	X		X	X		
Discoverability of vulnerability for the adversary (e.g., DREAD or CVSS discoverability score)	Pd	N	N	O	M	D				X	X	X	X				
Number of threat models created and analyzed	Pd, Pc	S	N	ET	M	D	X	X					X	X			
Number of threats identified during threat modeling	Pd, Pc, Re	S	N	ET	M	D	X	X					X	X			
Number of vulnerabilities found (e.g., during test or during operation)	Pd, Pc, Re	D	N	TT	M	D		X	X	X	X	X		X			
Effectiveness / Rigor: Protection, Detection, Response																	
Strength of mechanism	Pd	N	P	O	A	C	X	X	X	X			X	X	X		
Resilience (e.g., ratio between successful and unsuccessful attacks)	Pd	D	N	TT	M	D				X		X		X			X
Number of vulnerabilities identified (e.g., during test or during operation)	Pd, Pc, Re	D	N	TT	M	D		X	X	X	X	X		X			X
Vulnerability score (e.g., CVSS Base Score) of identified vulnerabilities	Pd	N	N	O	A	D				X		X		X			
Percentage of incidents that exploited vulnerabilities with known solutions (Protection quality)	Pc, Pd	D	U	TT	M	C						X		X	X	X	

Name	Target	Target Changeability	Measure Changeability	Measurement Frequency	Measure / Assess	Descriptive / Constructive	Usage in the Life Cycle						Measurement Goal				
							Requirements	Design	Implementation	Test and Inspection	Deployment	Operation	Security Requirement	Identification	Mitigation	Compliance / Coverage	Return on Investment
Percentage of systems attacked by exploits of known vulnerabilities (Response quality)	Pd	D	U	TT	M	D						X		X		X	
Number/frequency of incidents (of different types)	Pd, Pc, Re	D	N	TT	M	D						X					X
Meantime to compromise (e.g., relative to detection/reaction time)	Pd	D	N	TT	A	D				X	X	X		X			
Vulnerability discovery rate (during development)	Pd, Pc	D	N	TT	M	D				X							X
Meantime to detect ongoing incidents (detection delay)	Pd, Pc, Re	D	N	TT	M	D						X					X
Detection efficiency (e.g., number of false alarms)	Pd, Pc, Re	D	N	TT	M	D						X					X
Average frequency of log inspection	Pc	D	N	TT	M	D						X	X	X			
Remediation delay (e.g., meantime to respond to security incidents)	Pc, Re	D	N	TT	M	D			X			X					X
Vulnerability elimination rate (of known vulnerabilities)	Pc, Re	D	N	TT	M	D			X			X					X

4.3 Remarks

Our categorization of indicators reveals the following observations:

- There are more than twice as many measurement targets that change dynamically than there are targets that change statically. Nearly all measures that are applicable to dynamically changing targets are used in the operation phase.
- Most of the analyzed measures are not assumed to change over time. This means that the scale of the measure and the interpretation of the results are considered stable. For some measures it is hardly possible to avoid the changeability as they depend on environmental changes such as new threats or attacks.
- The measurement frequency strongly depends on the changeability of target and measure. A frequency matrix is shown in Table 6. If target and measure do not change at all, only one initial measurement is necessary. Regular changes in the target or measure obviously require repeated measurements. The interval between measurements is determined by the frequency and nature of target and measure changes. It is also possible that the measurement with a fixed or predictable measure is triggered by specific events such as the patching of a system (static change). Such trigger points are rarely described in the literature (NIST 800-55 [CSS+08] is one of the few exceptions). It is recommended that authors of measures define adequate measurement intervals or trigger points in the measure descriptions.

Table 6: Measurement Frequency

Measurement Frequency		**Target Changeability**		
		No Changes	Static Changes	Dynamic Changes
Measure Changeability	No Changes	Once	Event Triggered	Event / Time Triggered
	Predicable Changes	Time Triggered	Event / Time Triggered	Event / Time Triggered
	Unpredictable Changes	Event / Time Triggered	Event / Time Triggered	Event / Time Triggered

- One third of the categorized indicators have the ability to assess targets beyond only measuring them (for our definition of »assessing« as compared to »measuring« cf. Figure 1 on page 11). It is not trivial to use the results of a measure that does not have a clear scope and a clear baseline as we define it for a metric in Chapter 2.
- More than half of the investigated measures are only descriptive. That is, they help to reveal security issues, but they provide little insight in how the security problem could be mitigated. Indicators that are constructive do provide the necessary information to change the crucial security parameters. Especially the mitigation of security flaws in the development phase of a product would be of great benefit, but unfortunately only few indicators support mitigation in early lifecycle phases.
- Security criteria catalogs and security best practice catalogs are the most established approaches to determine security needs or security properties of products and processes. Based on such catalogs, coverage and compliance measures can be easily derived.

In summary, our categorization shows that compared to established metrological standards, the art of security measurements is still considerably lagging behind: Most indicators and their measurement methods are only vaguely defined, their applicability is restricted, and their discriminatory or predictive power is quite limited. In Chapter 5, we shall have a closer look into these deficiencies and their underlying reasons.

5 Conclusion

This report analyzes and classifies 44 papers[1] that describe — in the broadest sense — security indicators. Some topics were deliberately excluded from this survey, such as indicators for intrusion detection mechanisms used in network security or for security modeling approaches. Although this is a small amount of publications compared to the available body of literature on the subject, this report provides a valid account on the overall situation in research on security indicators.

Several types of indicators that measure security were identified during this research. They can be divided in indicators that measure

- the *coverage* of standards or policy requirements (compliance) or the fraction of systems that satisfy a given requirement (target coverage)
- the expected *costs* for a successful attack, for taking countermeasures, or for loss and damage caused by security incidents
- the *probability* of being vulnerable to a threat or suffering an attack attempt
- the *effectiveness and rigor* of provisions to protect from, detect, or respond to security incidents

From the above indicators, an additional indicator class, the *return on security investment*, can be derived as an overarching effectiveness measure.

Security indicators and measures can effectively support the sustainable improvement of IT security. Measures such as Key Performance Indicators (KPI), as they are described in ITIL [itSMF07], are already well-established; they can help enterprises to obtain an overview of the security status of their organization and their IT systems. But our survey showed that we still lack adequate methods to devise, determine, analyze, and enhance the security indicators and measures themselves.

5.1 A Critical Review of Existing Security Indicators

This state-of-the-art report reveals that security specification, measurement and improvement are not yet mastered satisfactorily. During the literature review

[1] Significantly more papers have been evaluated for this study, among them several surveys on related subjects. We intensively read more than 120 publications and skimmed many more, but only 44 approaches were finally selected as most significant and most representative for our topic. Due to this selection process the number of references provided for an individual security indicator does not necessarily correlate with its importance and pervasiveness.

and the subsequent classification of approaches and indicators, we identified a lot of shortcomings in current approaches. No structured methodology for defining security metrics has been found, and not a single measurement completely matches our definition of a security metric, as it is stated in Chapter 2. Below we summarize and categorize our critique on currently available security indicators.

5.1.1 Measurement Scope, Relevance, and Significance

The measurement scope of an indicator is often addressed only superficially. For example, we know what »80 percent of systems are equipped with virus protection« literally means from a technical point of view, but it is hard to interpret this measure in terms of the effects on »loss expectation per year« or similar measures that are closer to »perceived actual security«. It is not trivial to use the results of such a measure that does not have a clear scope as we define it for a metric in Chapter 2.

A related problem is that there are hardly any empirical results that confirm the predictive power of security indicators. On the contrary, many indicators seem to be quite weak in prediction, and they correlate only weakly to actual security properties. While, for example, many indicators are able to determine the rate of compliance to established standards or best practice catalogs, hardly any indicator can serve as a reliable predictor for the expected security quality of a system in operation. Attempts to predict the »security in the field« seem to be poorly supported by existing indicators.

The often non-existing mapping of security metrics and measurements to concrete goals is another problem. What is the true benefit of improving the rating of a security indicator? What type of security incidents is it expected to prevent? Questions like these are often unanswered, especially in scientific papers. A positive example is NIST 800-55 [CSS+08]. For all its proposed metrics, the NIST guideline answers these questions satisfactorily.

Thus, most publications lack a discussion of the significance of the proposed metrics for actual »security in operation«. Better evaluation of the indicators' significance and more documented practical experience in applying the indicators to software, systems, or security processes are required before security can be reliably predicted, controlled, enforced, or proved with indicators.

The degree of significance of the proposed security metrics is also correlated with the metric types:

- Maturity level metrics are mainly described in standards. They rely on the security knowledge of security experts and seem to be well accepted and applied in industry. This expert consensus suggests a high significance.
- Seven of the analyzed publications propose cardinality metrics. All in all we found several hundred metrics that match the »number of …« or »per-

centage of …« schemas. The significance of these metrics strongly depends on the ability to interpret the measured values, which requires a clear baseline and a high objectivity of the metrics. In many cases, these requirements are not met.

- More complex formulas are introduced as security metrics in ten publications. It is essential for metrics with many parameters that they are well defined and objective, which they are not in most publications. The indicator value can only be significant if the determination of all parameter values has low latitude.

5.1.2 Theoretical Foundation and Empirical Validation

Hardly any indicator has a solid theoretical foundation, that is, an underlying security model that would explain the cause-and-effect relation or at least the correlation between the indicator value and the degree of security obtained. In most cases, not even empirical evidence in support of the claimed correlation is available. This lack of justification is further substantiated by the fact that some proposed indicators even are based on dubious, apparently incorrect theoretical foundations such as calculating mean values over measurement scales that lack the interval property.

Most security metrics and measurements have not been applied and tested adequately in a real life case study. They lack a solid validation. For example, security processes can improve the overall security in an organization, depending on the organization's level of process compliance. But a proper methodology for evaluating the correlation between the degree of process compliance and the actual security level is missing in the literature. Our finding is in line with Verendel's earlier results on the subclass of metrics for operational security; in [Ver09] he concluded: »Quantified Security is a Weak Hypothesis«.

Thus, the main focus of future research should not be put on the development of yet another approach, but rather on a convincing theoretical foundation, empirical evaluation, and improvement of existing approaches in respect to their impact on actual security in system operation.

5.1.3 Reproducibility and Objectivity

Many security metrics lack objectivity because the parameters and their measurement procedure are poorly described. In fact, it seems that many parameter values must be chosen subjectively. Consequently, the values needed to apply the metric may vary considerably, depending on the assessor’s understanding of the parameters and his momentary disposition. This subjectivity makes it difficult to reproduce the same measurement or to obtain consistent, repeatable assessment results. As a consequence, monitoring a system over time — let alone comparing measurements across organizational boundaries — is severely limited.

The inherent uncertainty of some seemingly objective measurements is problematic. For example, some security requirements are hard to verify after implementation because their definition is either too imprecise or too high-level. To measure »the fraction of all systems that have a given security-related property« we first have to determine the set of all systems that should have the required property. Let us assume, for instance, that computer hard drives should be encrypted. Should this apply to all computers, server devices as well as desktops, or only to mobile devices? Should we include smartphones (although — strictly speaking — they do not have a hard drive)? And how can we gain a consistent snapshot if a certain fraction of mobile devices is constantly out of reach for assessment?

5.1.4 Well-defined Parameters and Guidance in the Measurement Procedure

Many security metrics lack concrete measuring, an adequate description of the measure's scale, and reference values for comparison. Especially for a subjective indicator it is essential to specify a value scale together with guidelines for selecting a suitable value for the evaluation target.

Ideally a baseline or a clear description how to define a baseline should be provided by the metric developer. The baseline should specify target or reference values, respectively, thresholds, acceptance criteria, and the expected degree of coverage. These reference values can clarify the significance of a measured value. They help to estimate the effort that is necessary to improve the target and provide insight into the overall security gained by reaching an optimal indicator value.

Even those indicators that define a clear metric and scale for measurement typically provide poor interpretation concerning questions such as: If we raised the indicator value by 10 percent, how much security improvement would be gained? What would be a reasonable baseline value for the indicator in the context of our specific enterprise? How could we translate security goals or requirements into corresponding indicator values?

The usual approach is to apply metrics repeatedly on different versions of the same system and to compare the results. This reveals a security trend, but says little about the absolute level of security that is achieved.

Moreover, most works lack a concept for the aggregation of individual measures to an overall security metric. However, as security is an inherently global property of a system, it is unlikely than any local indicator (or any small collection of local indicators) is ultimately able to convey a suitable representation of the security status of a system. Therefore, integrating a spectrum of indicator values into an aggregated security measure would be desirable, but has not been solved satisfactorily yet.

5.1.5 Mitigation of Security Flaws

Many security indicators help to *reveal* security issues, but they provide little insight in how the measurement result can be positively influenced, that is, how the security problem could be mitigated. More than half of the investigated measures are only descriptive (cf. Table 5, p. 36). Indicators that are constructive provide the necessary information to change the crucial security parameters.

In our opinion, the identification of security flaws should always go hand in hand with proposals for remediation, or at least with best practices that result in measurable security improvements. In combination, measurement and improvement abilities could harden the security of systems efficiently because measurement helps to concentrate improvement effort where it is most needed, promoting issue-related security enhancements. Most measurement methods provide no direct assistance in security improvements. Hence, they leave security improving reactions without guidance.

Especially in the development phase of a product, the mitigation of security flaws would be highly beneficial, but unfortunately only few indicators support mitigation in early lifecycle phases.

5.1.6 Tool Support

Tools can be effective for automating security measurement and documentation, for guiding through security validation processes, and for improving the scalability of the assessment method to large and complex evaluation targets. Therefore the development of tools could strongly benefit the application of many security indicators.

Only six methods we surveyed have tool support. Tools are only available for methods with a generic context and a focus on the target »product«. Available tools for static analysis operate mostly at a syntactic level, with very limited capabilities in respect to security logic. Moreover, proposed tools for static analysis are plagued by false positives.

5.1.7 Applicability to Software

There are insufficient security metrics at code and application level. Most approaches address processes or the system at a high level of abstraction. Few provide applicable security metrics for secure software engineering. That is, we lack static indicators that could predict the security »quality in use« of system architectures or source code in advance of deployment, or that could demonstrate the correctness of implemented security functionality.

The significance of most of the proposed software metrics — i.e., their correlation with actual security — has not been soundly analyzed, demonstrated or

proven, or their parameters are imprecisely defined, leaving too much room for interpretation. In fact, no sophisticated security metrics catalog for software security measurements has been found that contains security metrics as well as reference values for interpreting the results.

There are two main reasons for this lack of applicable, meaningful software security metrics. Firstly, software is in most cases highly complex. Therefore it is far from trivial to find the significant locations for measurement. Secondly, software can contain a large diversity of security mechanisms. There are a lot of security frameworks, libraries, built-in security functions for several programming languages, and recognizable standard security solutions or security design patterns. In addition, security functions could be proprietary and therefore hard to identify — or of unknown implementation quality. Because of these problems, it is challenging to predefine measurement methods or metrics for software. It is even more challenging to automate security measurement under these circumstances.

5.1.8 Life Cycle Coverage

The different life cycle phases are covered quite unevenly by the known security indicators. Table 5 on page 36 reveals that the operation phase is best covered, while for the early phases of development — i.e., design, implementation, and test — relatively few security indicators are available. Among the »early« indicators, the majority addresses process rather than product properties.

Thus, there is an urgent need for security metrics that could support product quality assurance in early stages of development.

5.1.9 Cost Indicators

Many proposed indicators rely rather naively on cost estimations. However, measuring costs, damage, or losses can be quite difficult, even in retrospective. For example, it is straightforward to suggest determining »the losses in the reputation and goodwill«, but without a detailed specification of the measurement procedure this metric is meaningless because the actual costs of reputation loss can turn out to be quite incalculable. Similarly, the equation »expected loss = likelihood of incident times impact of incident« may be mathematically sound, but it is still meaningless as long as we neither know likelihood nor impact with reasonable accuracy.

In particular, using »attack costs« as an indicator for the likelihood of an attack can be quite deceptive. Firstly, available technology for attacking may change rapidly, so that attacks turn out to be cheaper than expected. Secondly, an adversary may shift the costs to a third party, for example, by subverting some Internet servers and installing a bot net that provides massive computing power virtually for free — at least from the adversary's perspective. Therefore, even considerable costs need not deter an attacker. An even easier approach is to

rent capacities in a commercial cloud for an attack. In January 2011 Thomas Roth demonstrated how to use the Amazon Elastic Compute Cloud (EC2) to crack WPA keys [Kni11].

5.1.10 Return on Investment

Our literature research revealed several methods for measuring the return on security investment (ROSI). ROSI metrics can be seen as indicators as they give feedback on security improvement after investing money. The problem with these metrics is that the parameters of all ROSI equations are meant to be objective and numerical, but in most cases they are insufficiently defined. The methods do not describe how to obtain the values for these parameters. Using imprecise and highly subjective parameters, the result of the ROSI calculations will be imprecise and subjective, too.

5.1.11 Extrapolation of Security Measurements

Security metrics often ignore — or underestimate — the constant variability of security threats and security goals. If an attack was popular last year, this does not necessarily mean that it will occur as frequently in the next few months; that our firewall was »good enough« in the past does not indicate that we are still safe in the future. Both attack and defense potentials are subject to constant change, and so are our security needs. Therefore, we cannot simply transfer our past experience (e.g., likelihood and impact of security incidents) to the future, as many methods seem to imply.

Some indicators implicitly or explicitly rely on extrapolation. For example, the frequency of past security incidents is taken as a predictor for the expected likelihood of future incidents. However, in the realm of security extrapolation has to be exercised with caution. Motivation, targets, and potentials for attack are subject to rapid, often unpredictable change. The same holds for the prevalent information and software technology, opening IT operations to an unforeseeable sequence of newly discovered security gaps.

5.1.12 Maturity of Security Indicators

Security requirements definition seems momentarily to be the most mature field in security research. One indication is the degree of standardization, as witnessed by our report: all described approaches that belong to the functionality class »Security Requirements Definition« (cf. Table 2, p. 15) are standardized. In fact, collecting, classifying and standardizing all kinds of security requirements or security criteria has a long tradition, as exemplified by the *Trusted Computer System Evaluation Criteria* (the so-called »orange book«, 1983) and the *Common Criteria for Information Technology Security Evaluation* (introduced in 1998).

Apart from security evaluation criteria, there are a few standards and regularly used approaches in the field of security process definition and improvement. However, there are still few mature approaches in the field of security *measurement*. The lack of established or even standardized security metrics reflects the fundamental conceptual difficulties of security quantification: quantitative security assessment is still an unsolved research problem!

As the field of security requirements definition is more advanced than security measurement, the focus should be on measurement in future research. It would be desirable to have criteria catalogs that do not only specify security requirements, but also attach metrics (or measurable conditions that are key to success) to each requirement that measure the effectiveness and efficiency of their implementation.

5.2 Towards Better Security Indicators

As we saw in Section 5.1, available security indicators have significant deficiencies. Despite considerable research on the topic, finding, rating, and eliminating security defects persists to be a human-centric art that is hard to automate.

For a major breakthrough in security assurance, substantial progress in security measurement is required. More specifically, we need more reliable and more meaningful security indicators. Ideally, such indicators should meet the following requirements:

- The indicator should have a well-defined assessment scope that addresses a specific security goal.
- The indicator should provide a rationale, that is, a theoretical foundation or empirical evidence proving its predictive power and its significance for security evaluation.
- The indicator should be based on objective parameters and on a reproducible measurement procedure.
- The indicator should be quantitatively measurable on a well-defined measurement scale with established baseline and target values.
- The indicator should provide guidance for the interpretation of the measurement results.
- The indicator should not only signal unspecific security deficiencies; where possible, it should constructively point out improvement potential and opportunities to close existing security gaps.
- The indicator should be applicable to large and complex evaluation targets; to this end, tool-support is strongly recommended.

We are aware that not all of these properties equally apply to each proposed security indicator; still the conceptual design of new indicators should strive for this ideal.

Besides the above-mentioned characteristics of individual indicators, a more balanced coverage of the whole system life-cycle is also desirable. More specifically, we need

- Security indicators for design and implementation *products* as well as for development, deployment, and operation *processes*; more and better indicators for software and software architectures would be of particular interest
- Security indicators for the *early lifecycle phases* (i.e., design, implementation, and test) as well as for the *late phases* (i.e., deployment and operation).

Today there is an imbalance towards process metrics and late lifecycle phases.

References

[AD03] Muhammad Al-Humaigani, Derrek B. Dunn: A Model of Return on Investment for Information Systems Security. Proceedings of the 46th IEEE international midwest symposium on circuits & systems, Vol. 1, pp. 483–485, Cairo, Egypt, 2003

[ASD00] Assistant Secretary of Defense for Command, Control, Communications, and Intelligence: Department of Defence Information Technology Security Certification and Accreditation Process (DITSCAP). 2000, http://biotech.law.lsu.edu/blaw/dodd/corres/pdf/85101m_0700/p85101m.pdf (last access: January 03, 2012).

[BHW06] Pauline Bowen, Joan Hash, Mark Wilson: NIST Special Publication 800-100. Information Security Handbook: A Guide for Managers. National Institute of Standards and Technology, 2006, http://csrc.nist.gov/publications/nistpubs/800-100/SP800-100-Mar07-2007.pdf (last access: January 03, 2012).

[BM07] Wayne Boyer, Miles McQueen: Ideal Based Cyber Security Technical Metrics for Control Systems. Proceedings of the Second International Workshop on Critical Information Infrastructures Security, pp. 246–261, Malaga, Spain, 2007.

[BMB+08] Nadya Bartol, Bob Martin, Sean Barnum, et al.: Practical Measurement Framework for Software Assurance and Information Security (DRAFT). 2008, http://www.psmsc.com/Downloads/TechnologyPapers/SwA%20Measurement%2010-08-08.pdf (last access: January 03, 2012).

[Bru06] Jochen Brunnstein: ITIL Security Management realisieren. Vieweg & Sohn Verlag, Wiesbaden, 2006, ISBN 3-8348-0165-8.

[BSI05] Bundesamt für Sicherheit in der Informationstechnik (BSI): IT Grundschutz Catalogue. 2005, https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Grundschutz/download/it-grundschutz-kataloge_2005_pdf_en_zip.zip?__blob=publicationFile (last access: January 03, 2012).

[BSI09] Bundesamt für Sicherheit in der Informationstechnik (BSI): IT Grundschutzkataloge, 11. Ergänzungslieferung – November 2009, https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Grundschutz/Download/it-grundschutz-kataloge_2009_EL11_de.pdf?__blob=publicationFile (last access: January 03, 2012).

[CC09] Common Criteria for Information Technology Security Evaluation: Part 1: Introduction and general model. Version 3.1, 2009, http://www.commoncriteriaportal.org/files/ccfiles/CCPART1V3.1R3.pdf (last access: January 03, 2012).

[CC09a] Common Criteria for Information Technology Security Evaluation: Part 2: Security functional components. Version 3.1, 2009, http://www.commoncriteriaportal.org/files/ccfiles/CCPART2V3.1R3.pdf (last access: January 03, 2012).

[CC09b] Common Criteria for Information Technology Security Evaluation: Part 3: Security assurance components. Version 3.1, 2009, http://www.commoncriteriaportal.org/files/ccfiles/CCPART3V3.1R3.pdf (last access: January 03, 2012).

[CISWG04] Corporate Information Security Working Group: Report of the Best Practices and Metrics Teams. 2004, http://net.educause.edu/ir/library/pdf/CSD3661.pdf (last access: January 03, 2012).

[CMU03] Carnegie Mellon University: Security Software Engineering Capability Maturity Model SSE-CMM. Model Description Document. Version 3.0, 2003, www.sse-cmm.org/docs/ssecmmv3final.pdf (last access: January 03, 2012).

[CSS+08] Elizabeth Chew, Marianne Swanson, Kevin Stine, Nadya Bartol, Anthony Brown, Will Robinson: Performance Measurement Guide for Information Security. NIST Special Publication 800-55 Revision 1, National Institute of Standards and Technology, 2008, http://csrc.nist.gov/publications/nistpubs/800-55-Rev1/SP800-55-rev1.pdf (last access: January 03, 2012).

[CW07] Brian Chess, Jacob West: Secure Programming with Static Analysis. Software Security Series, Addison-Wesley, Upper Saddle River, 2007, ISBN 0-321-42477-8.

[Dav09] Mary Ann Davidson: The Good, The Bad, And The Ugly: Stepping on the Security Scale.. Proceedings of the Annual Computer Security Applications Conference, Honolulu, Hawaii, USA, pp. 187–195, 2009, www.acsac.org/2009/program/keynotes/davidson.pdf (last access: January 03, 2012).

[Die09] DiePresse.com: Hacker-Attacke auf Südkorea: Österreich unter Verdacht (German), http://diepresse.com/home/politik/aussenpolitik/493971/index.do?_vl_backlink=/home/politik/aussenpolitik/index.do (last accessed: January 03, 2012).

[DMS07] Mark Dowd, John McDonald, Justin Schuh: The Art of Software Security Assessment. Identifying and Preventing Software Vulnerabilities. Addison-Wesley, Upper Saddle River, 2007, ISBN 0-321-44442-6.

[DSS+09] Chad Dougherty, Kirk Sayre, Robert C. Seacord, David Svoboda, Kazuya Togashi: Secure Design Patterns. Technical Report CMU/SEI-2009-TR-010 ESC-TR-2009-010, 2009, www.sei.cmu.edu/reports/09tr010.pdf (last accessed: January 03, 2012).

[EHM+09] Frank Elberzhager, Jens Heidrich, Michael Kläs, Constanza Lampasona, Jürgen Münch, Adam Trendowicz: Quamoco — Map of Existing Quality Models. Fraunhofer IESE, IESE-Report No. 107.09/E, 2009.

[EW05] Shelby Evans, James Wallner: Risk-based Security Engineering through the Eyes of the Adversary. Proceedings of the Sixth Systems, Man and Cybernetics Information Assurance Workshop, pp. 158–166, West Point, New York, 2005.

[GoI10] Government of India - Department of Information Technology - Ministry of Communications and Information Technology: E Governance Security Standards Framework: An Approach Paper. 2010, http://egovstandards.gov.in/guidelines/guidelines-for-information-security/eSAFE-Framework-ApproachPaper.pdf (last accessed: January 03, 2012).

[GoI10a] Government of India - Department of Information Technology - Ministry of Communications and Information Technology: E Security Assurance Framework: Catalog of Security Controls eSAFE GD 200. 2010, http://egovstandards.gov.in/guidelines/guidelines-for-information-security/eSAFE-GD200-CatalogOfSecurityControls.pdf (last accessed: January 03, 2012).

[GWM+07] Karen Mercedes Goertzel, Theodore Winograd, Holly Lynne McKinley, Lyndon J. Oh, Michael Colon, Thomas McGibbon, Elaine Fedchak, and Robert Vienneau: Software Security Assurance. Joint State-of-the Art Report, Information Assurance Technology Analysis Center (IATAC), Data and Analysis Center for Software (DACS), 2007, http://iac.dtic.mil/iatac/download/security.pdf (last accessed: January 03, 2012).

[Her10] Pete Herzog: OSSTMM 3 — Open-Source Security Testing Methodology Manual. ISECOM, 2010, http://www.isecom.org/mirror/OSSTMM.3.pdf (last accessed: January 03, 2012).

[Her07] Debra S. Herrmann: Complete Guide to Security and Privacy Metrics. Auerbach Publications, Boca Raton, 2007, ISBN: 978-0-8493-5402-1.

[HL03] Michael Howard, David LeBlanc: Writing Secure Code. Practical strategies and techniques for secure application coding in a networked world. Microsoft Press, Redmond, 2003, ISBN 0-7356-1722-8.

[HYY09] Rui Huang, Danfeng Yan, Fangchun Yang: Research of Security Metric Architecture for Next Generation Networks. IEEE International Conference on Network Infrastructure and Digital Content, pp. 207–212, Beijing, 2009.

[ISSA04] Information Systems Security Association (ISSA): Generally Accepted Information Security Principles (GAISP). Version 3.0, 2004, http://all.net/books/standards/GAISP-v30.pdf (last accessed: January 03, 2012).

[ITGI07] IT Governance Institute: COBIT 4.1 Excerpt. 2007, http://www.isaca.org/Knowledge-Center/cobit/Documents/COBIT4.pdf (last accessed: January 03, 2012).

[itSMF07] itSMF: An Introductory Overview of ITIL V3. Version 1.0, 2007, http://www.itsmfi.org/files/itSMF_ITILV3_Intro_Overview_0.pdf (last accessed: January 03, 2012).

[Kni11] Terry Relph-Knight: Cracking WPA keys in the cloud. The H security, http://www.h-online.com/security/news/item/Cracking-WPA-keys-in-the-cloud-1168636.html (last accessed: January 03, 2012).

[KRE09] Brian Krebs: Payment Processor Breach May Be Largest Ever. The Washington Post, http://voices.washingtonpost.com/securityfix/2009/01/payment_processor_breach_may_b.html?hpid=topnews (last accessed: January 03, 2012).

[LB08] David John Leversage, Eric James Byres: Estimating a System's Mean Time-to-Compromise. IEEE Security and Privacy 6(1), pp. 52–60, 2008.

[LPF+08] Volkmar Lotz, Emmanuel Pigout, Dr. Peter M. Fischer, Prof. Dr. Donald Kossmann, Prof. Dr. Fabio Massacci, Dr. Alexander Pretschner: Towards Systematic Achievement of Compliance in Service-Oriented Architectures: The MASTER Approach. Wirtschaftsinformatik 50(5), pp. 383–391, 2008.

[LZR+09] Xiaoli Lin, Pavol Zavarsky, Ron Ruhl, Dale Lindskog: Threat Modeling for CSRF Attacks. SIAM Conference on Computational Science and Engineering (CSE '09), pp. 486–491, Miami, Florida, USA, 2009.

[Mic05] Microsoft: patterns & practices Security Checklists Index, http://msdn.microsoft.com/en-us/library/ms998392.aspx (last accessed: January 03, 2012).

[Mic10] Microsoft: Security Development Lifecycle. Simplified Implementation of the Microsoft SDL. 2010, http://www.child-traffick-ing.info/upload/Files/Simplified%20Implementation%20of%20the%20SDL.pdf (last accessed: January 03, 2012).

[Mic10a] Microsoft: Security Development Lifecycle. Version 5.0, March 31, 2010, http://www.microsoft.com/downloads/details.aspx?FamilyID=7d8e6144-8276-4a62-a4c8-7af77c06b7ac&displaylang=en (last accessed: January 03, 2012).

[Mit11] MITRE: Common Weakness Scoring System (CWSS™). http://cwe.mitre.org/cwss/index.html (last accessed: January 03, 2012).

[MKC+08] Matteo Meucci, Eoin Keary, Daniel Cuthbert, et al.: OWASP Testing Guide. Version 3.0, 2008, https://www.owasp.org/images/5/56/OWASP_Testing_Guide_v3.pdf (last accessed: January 03, 2012).

[MSR07] Peter Mell, Karen Scarfone, Sasha Romanosky: A Complete Guide to the Common Vulnerability Scoring System. Version 2.0, 2007, http://www.first.org/cvss/cvss-guide.pdf (last accessed: January 03, 2012).

[MTM+07] Pratyusa K. Manadhata, Kymie M. C. Tan, Roy A. Maxion, Jeannette M. Wing: An Approach to Measuring A System's Attack Surface. Computer Science Technical Report CMU-CS-07-146, 2007, http://reports-archive.adm.cs.cmu.edu/anon/2007/CMU-CS-07-146.pdf (last accessed: January 03, 2012).

[MW05] Pratyusa Manadhata, Jeannette M. Wing: An Attack Surface Metric. Computer Science Technical Report CMU-CS-05-155, 2005, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.60.6977&rep=rep1&type=pdf (last accessed: January 03, 2012).

[NP07] Elisabeth A. Nichols, Gunnar Peterson: A Metrics Framework to Drive Application Security Improvement. Security & Privacy, IEEE, Vol. 5, No. 2, pp. 88–91, 2007.

[Olb04] Alfred Olbrich: ITIL kompakt und verständlich. Vieweg & Sohn Verlag, Wiesbaden, 2004, ISBN 3-528-15892-1.

[OSF10] Open Security Foundation: Data Loss Database – March, 2010 report. http://datalossdb.org/monthly_reports/2010/dataloss-2010-3.pdf (last accessed: January 03, 2012).

[OSF10a] Open Security Foundation: Data Loss Database – 2009 yearly report. http://datalossdb.org/yearly_reports/dataloss-2009.pdf (last accessed: January 03, 2012).

[OWA05] OWASP — A Guide to Building Secure Web Applications and Web Services. 2005, http://downloads.sourceforge.net/project/owasp/Guide/2.0.1/OWASPGuide2.0.1.pdf (last accessed: January 03, 2012).

[OWA09] OWASP — OWASP Application Security Verification Standard. 2009, http://www.owasp.org/images/4/4e/OWASP_ASVS_2009_Web_App_Std_Release.pdf (last accessed: January 03, 2012).

[PCI10] PCI Security Standards Council LLC: PCI DSS Requirements and Security Assessment Procedures. v2.0, 2010, https://www.pcisecuritystandards.org/documents/pa-dss_v2.pdf (last accessed: January 03, 2012).

[Pet09] Gunnar Peterson: Service-Oriented Security Indications for Use. Security & Privacy, IEEE, Vol. 7, No. 2, pp. 91–93, 2009.

[PF08] P. Parrend, S. Frenot: Security benchmarks of OSGi platforms: toward Hardened OSGi. Journal Software – Practice & Experience, Vol. 39, No. 5, pp. 471–499, 2009.

[Pil10] Mary Pilon: Data Theft at Loan Firm Hits Borrowers. The Wall Street Journal. http://online.wsj.com/article/SB10001424052748704100604575146360834054370.html (last accessed: January 03, 2012).

[PJM08] Holger Peine, Marek Jawurek, Stefan Mandel: Security Goal Indicator Trees: A Model of Software Features that Supports Efficient Security Inspection. 11th IEEE High Assurance Systems Engineering Symposium, Nanjing, China, 2008.

[PPN06] Victor-Valeriu Patriciu, Iustin Priescu, Sebastian Nicolaescu: Security Metrics for Enterprise Information Systems. Journal of Applied Quantitative Methods, Vol. 1, No. 2, pp. 151–159, 2006, http://jaqm.ro/issues/volume-1,issue-2/pdfs/patriciu_priescu_nicolaescu.pdf (last accessed: January 03, 2012).

[Pur04] Steve A. Purser: Improving the ROI of the security management process. Computers & Security; Vol. 23, No. 7, pp. 542–546, 2004.

[RS12] Manuel Rudolph, Reinhard Schwarz: Security Indicators – A State of the Art Survey – Internal Report. Fraunhofer IESE, IESE-Report No. 001.12/E, 2012.

[RSP+09] Ron Ross, Gary Stoneburner, Esten Porter, et al.: NIST Special Publication 800-53 Revision 3. Recommended Security Controls for Federal Information Systems and Organizations. National Institute of Standards and Technology, 2009, http://csrc.nist.gov/publications/nistpubs/800-53-Rev3/sp800-53-rev3-final_updated-errata_05-01-2010.pdf (last accessed: January 03, 2012).

[RSS+10] Ron Ross, Gary Stoneburner, Terry Sherald, et al.: NIST Special Publication 800-53A Revision 1. Guide for Assessing the Security Controls in Federal Information Systems. National Institute of Standards and Technology, 2008, http://csrc.nist.gov/publications/nistpubs/800-53A-rev1/sp800-53A-rev1-final.pdf (last accessed: January 03, 2012).

[RR08] Julie J.C.H. Ryan, Daniel J. Ryan: Performance Metrics for Information Security Risk management. Security & Privacy, IEEE, Vol. 6, No. 5, pp. 38–44, 2008.

[Set06] Sekar Sethuraman: Framework for Measuring and Reporting Performance of Information Security Programs in Offshore Outsourcing. ISACA Journal, Vol. 6, pp. 1–8, 2006, http://www.isaca.org/Journal/Past-Issues/2006/volume-6/Documents/jopdf0606-framework-for-measuring.pdf (last accessed: January 03, 2012).

[SGF02] Gary Stoneburner, Alice Goguen, Alexis Feringa: NIST Special Publication 800-30 — Risk Management Guide for Information Technology Systems. National Institute of Standards and Technology, 2002, http://csrc.nist.gov/publications/nistpubs/800-30/sp800-30.pdf (last accessed: January 03, 2012).

[SH05] Martin Stoddard, Yacov Haimes: Process Control System Security Metrics — State of Practice. Institute for Information Infrastructure Protection (I3P), Research Report No. 1, 2005, http://stuweb.ee.mtu.edu/~ssmoily/section_4.pdf (last accessed: January 03, 2012).

[SHF04] Gary Stoneburner, Clark Hayden, Alexis Feringa: NIST Special Publication 800-27 Revision A — Engineering Principles for Information Technology Security (A Baseline for Achieving Security). National Institute of Standards and Technology, 2004, http://csrc.nist.gov/publications/nistpubs/800-27A/SP800-27-RevA.pdf (last accessed: January 03, 2012).

[Sin09] Ryan Singel: Probe Targets Archives' Handling of Data on 70 Million Vets. Wired Magazine, http://www.wired.com/threatlevel/2009/10/probe-targets-archives-handling-of-data-on-70-million-vets/ (last accessed: January 03, 2012)

[SMW+11] Yonghee Shin, Andrew Meneely, Laurie Williams, and Jason A. Osborne: Evaluating Complexity, Code Churn, and Developer Activity Metrics as Indicators of Software Vulnerabilities. IEEE Transaction on Software Engineering, Vol. 37, No. 6, pp. 772–787, 2011 DOI 10.1109/TSE.2010.81

[SNL06] Christopher Steel, Ramesh Nagappan, Ray Lai: core Security Patterns. Best Practices and Strategies for J2EE™, Web Services, and Identity Management. Pearson Education, Upper Saddle River, 2006, ISBN 0-13-146307-1

[Sow10] Aleksandra Sowa: Access Control Controls. <kes> Die Zeitschrift für Informationssicherheit, No. 1, 2010, pp.12–17.

[Sud09] John Sudworth: New 'cyber attacks' hit S Korea. BBC News, http://news.bbc.co.uk/2/hi/asia-pacific/8142282.stm (last accessed: January 03, 2012).

[T4A06] Trust4All Deliverable 1.6: Trust Framework and mechanisms. Version 1.0, 2006.

[Ver09] Vilhelm Verendel: Quantified Security is a Weak Hypothesis. A critical survey of results and assumptions. New Security Paradigms Workshop, Oxford, United Kingdom, 2009, http://www.nspw.org/papers/2009/nspw2009-verendel.pdf (last accessed: January 03, 2012).

[VFP04] Carlos Villarrubia, Eduardo Fernández-Medina, Mario Piattini: Towards a Classification of Security Metrics. Proceedings of the 2nd International Workshop on Security In Information Systems, pp. 342-350, Porto, Portugal, 2004.

[VGM+96] J. Voas, A. Ghosh, G. McGraw, F. Charron, E. Miller: Defining an Adaptive Software Security Metric from a Dynamic Software Failure Tolerance Measure. Proceedings of the Eleventh Annual Conference onComputer Assurance, Gaithersburg, Maryland, 1996.

[VHS03] Rayford B. Vaughn, Jr., Ronda Henning, Ambareen Siraj: Information Assurance Measures and Metrics - State of Practice and Proposed Taxonomy. Proceedings of 36th Hawaii International Conference on System Science, Big Island, Hawaii, USA, 2003.

[Vor09] David Voreacos: Hacker Agrees to Plead Guilty in Second Computer Data Theft. Bloomberg, http://www.bloomberg.com/apps/news?pid=20601110&sid=aE0.8o_7QcGc (last accessed: January 03, 2012).

[WNZ+07] Chris Wysopal, Lucas Nelson, Dino Dai Zovi, Elfriede Dustin: The Art of Software Security Testing. Identifying Software Security Flaws. Addison-Wesley, Upper Saddle River, 2007, ISBN 0-321-30486-1.

[WXZ07] Andy Ju An Wang, Min Xia, Fengwei Zhang: Metrics for Information Security Vulnerabilities. Proceedings of Intellectbase International Consortium, Volume 1, Atlanta, GA, USA, pp. 294–304, 2007.

Document Information

Title:	Security Indicators – A State of the Art Survey – Public Report
Date:	March 2012
Report:	043.12/E
Status:	Final
Distribution:	Public