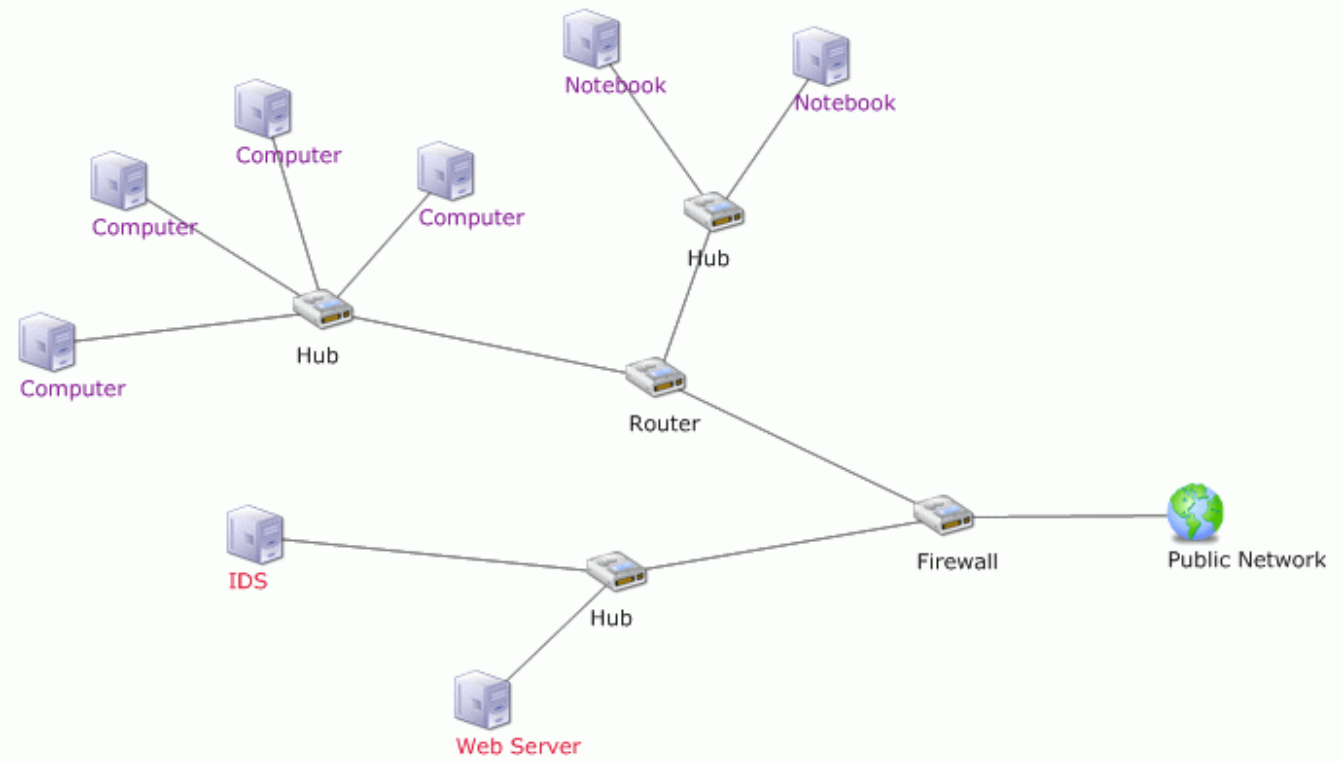
FOI-R--1468--SE
December 2004
ISSN 1650-1942
Scientific report

Jonas Hallberg, Amund Hunstad, Anders Bond, Mikael Peterson, Nils Påhlsson

System IT Security Assessment



Command and Control Systems
SE-581 11 Linköping

SWEDISH DEFENCE RESEARCH AGENCY

Command and Control Systems
P.O. Box 1165
SE-581 11 Linköping

FOI-R--1468--SE

December 2004

ISSN 1650-1942

Scientific report

Jonas Hallberg, Amund Hunstad, Anders Bond, Mikael Peterson, Nils Påhlsson

System IT Security Assessment

Issuing organization	Report number, ISRN	Report type
FOI – Swedish Defence Research Agency Command and Control Systems P.O. Box 1165 SE-581 11 Linköping	FOI-R--1468--SE	Scientific report
	Research area code 41 C4I	
	Month year December 2004	**Project no.** E7046
	Sub area code 41 C4I	
	Sub area code 2	

Author/s (editor/s)	Project manager
Jonas Hallberg Amund Hunstad Anders Bond Mikael Peterson Nils Påhlsson	Jonas Hallberg
	Approved by
	Sponsoring agency Swedish Armed Forces
	Scientifically and technically responsible

Report title

System IT Security Assessment

Abstract (not more than 200 words)

IT security is an issue of vital importance for all IT-based systems. As IT is penetrating the society, IT security becomes increasingly important. Unfortunately, IT security is intrinsically difficult to handle and motivate. Security assessment is a central ability in the striving for adequate levels of IT security in systems. In this report, an effort to enable system-wide IT security assessment is described. The presented results include:

- A study of current security evaluation methods.
- Terminology for the area of security assessment.
- A framework for system security assessment.
- A method for system security assessment.
- A framework for system component security assessment.
- A method for system component security assessment.

Keywords

IT security, security level, security metrics, component security assessment, system security assessment

Further bibliographic information	Language English

ISSN 1650-1942	Pages 86 p.
	Price acc. to pricelist

Utgivare Totalförsvarets Forskningsinstitut - FOI Ledningssystem Box 1165 581 11 Linköping	**Rapportnummer, ISRN** FOI-R--1468--SE	**Klassificering** Vetenskaplig rapport
	Forskningsområde 4. Ledning, informationsteknink och sensorer	
	Månad, år December 2004	**Projektnummer** E7046
	Delområde 41 Ledning med samband och telekom och IT-system	
	Delområde 2	

Författare/redaktör Jonas Hallberg Amund Hunstad Anders Bond Mikael Peterson Nils Påhlsson	**Projektledare** Jonas Hallberg
	Godkänd av
	Uppdragsgivare/kundbeteckning Försvarsmakten
	Tekniskt och/eller vetenskapligt ansvarig

Rapportens titel (i översättning)

Värdering av IT-säkerhet i system

Sammanfattning (högst 200 ord)

Ändamålsenlig IT-säkerhet är av stor vikt för alla IT-baserade system. Eftersom IT blir allt viktigare för alla delar av samhället, blir IT-säkerheten allt mer central. Tyvärr är IT-säkerhet svårt både att hantera och motivera. Förmågan att kunna värdera IT-säkerheten i system är kritisk för strävan att uppnå adekvata IT-säkerhetsnivåer. I denna rapport beskrivs ett arbete som syftar till att finna bra metoder för värdering av IT-säkerhet i system. Uppnådda resultat inkluderar:

- En studie av befintliga metoder för evaluering av IT-säkerhet i system.
- Terminologi för området värdering av IT-säkerhet.
- Ett ramverk för värdering av IT-säkerhet i system.
- En metod för värdering av IT-säkerhet i system.
- Ett ramverk för värdering av IT-säkerhet hos systemkomponenter.
- En metod för värdering av IT-säkerhet hos systemkomponenter.

Nyckelord

IT-säkerhet, säkerhetsnivåer, säkerhetsmetriker, värdering

Övriga bibliografiska uppgifter	**Språk** Engelska
ISSN 1650-1942	**Antal sidor:** 86 s.
Distribution enligt missiv	**Pris: Enligt prislista**

Contents

1. Introduction

IT security is an issue of vital importance for all IT-based systems. As IT is penetrating the society, a process which has merely started, IT security becomes increasingly important. Unfortunately, it is often difficult realize the need of security until it is too late. IT security is even more abstract than security in general and, thus, intrinsically difficult to handle and motivate.

In this report, an effort to enable system-wide IT security assessment is described. The text supposes knowledge of general IT security, which will not be explained here, since there already are several excellent sources, see for example (Gollmann, 1999; Anderson, 2001).

1.1 Motivation

As discussed in Chapter 2, researchers, industry, and IT security professionals argue for the necessity of being able to perform IT security assessments for information systems. Appropriate selection of security controls requires the ability to describe the security posture of information systems. Moreover, comprehensible descriptions of security postures and the effect of security controls motivate the introduction and use of these controls. This becomes increasingly important as information systems expand both in scope and business criticality.

Greenwald et al (2003) argue that despite the importance of current principles of IT security, these principles do not yield any way to assess the security of a system. To make real progress in the field of IT security there is a need to focus on three key areas:

- Development of better experimental techniques
- Development of better metrics of security
- Development of models with real predictive power

This, according to Greenwald et al (2003), is highly dependent on the establishment of a scientific foundation for future security research. Establishing such a scientific foundation is one of the major challenges in the field of IT security research.

1.2 Problem Formulation

The main issues that have to be resolved in order to facilitate efficient system security assessments are listed below.

- Mechanisms to specify the meaning of IT security for the particular system and situation are necessary. Security metrics, further discussed in Section 2.2 below, are fundamental for this process.
- The definition of security metrics is central in order to specify what is actually to be measured.
- An approach to system security assessment has to be selected. That is, will the system be observed and tested? Will the security functionality and structure of the system be considered?

This report targets these three main issues of system security assessment. In doing so, a structural approach to system security assessment is taken. Structural approaches to system security assessment require:

- modeling techniques capturing the security-relevant information of the structure of the system,
- a set of measurable security-related characteristics that can be mapped to specified security metrics,
- mechanisms to associate appropriate sub-sets of the security-related characteristics to system components.
- methods for assessment of the security strength of system components, regarding the associated set of characteristics, and
- methods aggregating the results for individual system components, possibly including other factors, to system-wide security measures.

This report targets also these five issues of structural system security assessment.

1.3 Contributions

The main results produced by the efforts described in this report are:

- A study of current security evaluation methods.
- Terminology for the area of security assessment.
- A framework for system security assessment.
- A method for system security assessment.
- A framework for system component security assessment.
- A method for system component security assessment.

To illustrate the use of the methods for system component and system security assessment, two software tools supporting the respective methods have been implemented.

1.4 Report Layout

In chapter 2, the issue of security assessment is discussed. In chapter 3, characteristics of and some specific approaches to system security assessment are treated. In chapter 4, a framework for structural system security assessment is introduced. In chapter 5, an approach to system component security assessment is discussed. In chapter 6, a framework for system component security assessment is introduced. In chapter 7, conclusions are drawn.

2. Security Assessment

The subject of security assessment is fairly large. Our interpretation is that a security assessment process should capture the relevant attributes of a system in order to be able to answer the question: *Is the security of the system adequate*? Inadequate security will prove costly, either through security breaches, in the case of to poor security, or expensive systems and administration and hampered organizations, in the case of too rigid security. There are several different aspects that have to be considered in order to answer the question.

- The meaning of security has to be clearly defined for the particular system and situation, in order for the assessment to emphasize the system characteristics considered relevant. This includes the definition of security metrics, which is further discussed in Section 2.2 below.

- The scope of the system has to be defined. This problem is two-fold. Firstly, the physical limits of the system to be assessed have to be specified. Secondly, different aspects of a system that can be included are technical, organizational, individual, operational, and contextual. To arrive at a complete picture of the security of an information system, all of the three aspects mentioned above have to be addressed. Specific methods may of course be focused on one of the aspects resulting in the need to complement the results with the results of other methods.

- Since security cannot be directly measured, other system properties have to be measured. These properties are either *factors* that affect the security level, such as the use of certain security controls, or *consequences* that are an effect of a certain *security level*, such as security vulnerabilities, see Figure 1. Examples of factors that correlate to the security level of an IT system are the presence of a security policy, the number of users, whether the network is connected to the Internet or not, the operation of firewalls, and the use of encryption. Examples of consequences that correlate to the security level of an IT system are the number of unauthorized retrievals of certain information in the past month and the number of successful attempts to withhold certain information the past year. Based on measured factors and/or consequences the security of a system can be assessed.

- The scope of the assessment process, which is further discussed in Section 2.1 below.

- The validity of the assessment, i.e., the relation between estimated and real security values. If an existing system is assessed, the system always has a set of real security values. The difficulty lies in establishing and representing these

values. As soon as systems are analyzed, models are used. Even if an automatic tool detecting all the desired properties of a system was available, the result would be a model and thus the resulting security values would be estimates derived from a modeled system.

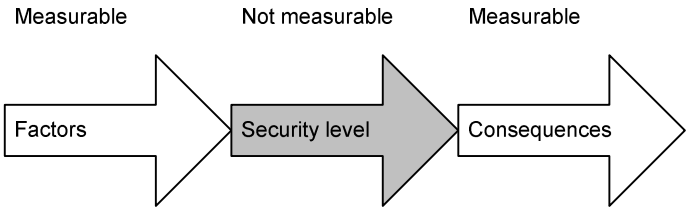


Figure 1: Measurable entities of an IT system in a security context.

2.1 Scope of the Assessment Process

This section is based on the chapter called *security measurement* in (Andersson et al, 2003).

The purpose of security assessment is to supply the risk management process with relevant, valid, and reliable data considering different security aspects of the system. To achieve this, the value of information has to be connected to the system components processing, storing, and transmitting the information and put into the context of the system. Thus, a fundamental input to risk management is results from the assessment of different security levels in systems and system components.

The system security level assessment can be divided into the tasks of assessing the security qualities of the entities constituting the system, that is the system securability, and how the system is operated. Consequently, the process of identifying, analyzing and describing system risks can be divided into securability assessment, security level assessment, context modeling, and risk level assessment, as illustrated in Figure 2. Each of these tasks requires input and generates results as illustrated in the figure.

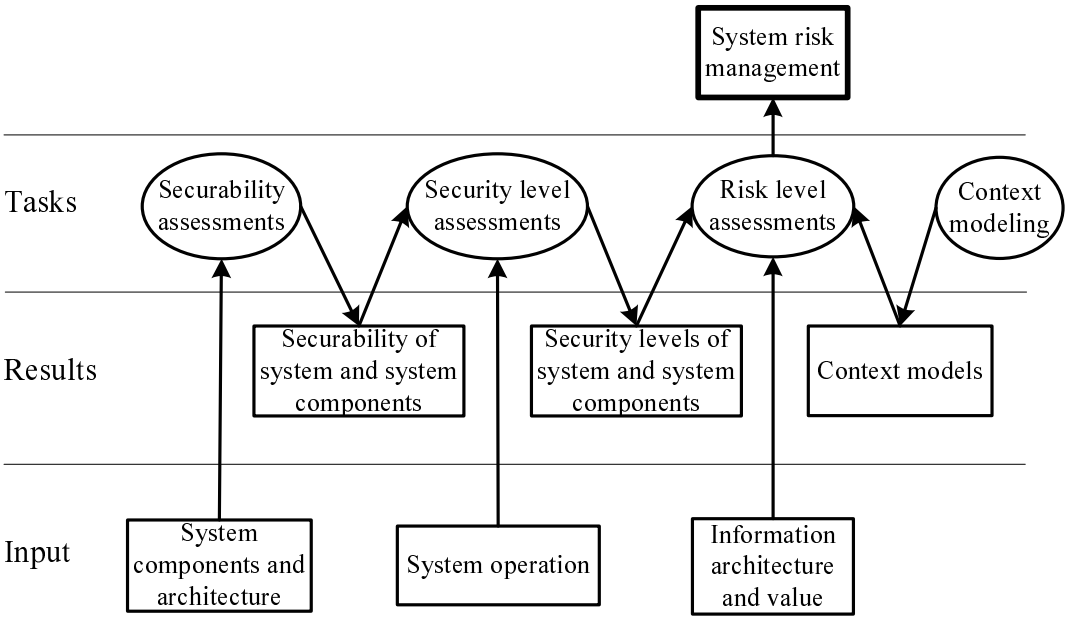


Figure 2: Securability and security level assessment are essential to efficient risk assessment.

In the remainder of this section, the security assessment tasks in Figure 2, namely securability, security level, and risk level assessment, and the related terms security value and security strength are discussed.

> **Securability**: The goal of designing for securability is that systems can be secured to a required level during operation (Hunstad and Hallberg, 2002). Thus, securability assessment aims at evaluating the strengths and weaknesses of security mechanisms considering technical, organizational, and individual aspects. Consequently, securability is assessed pre-operational or during operation ignoring the actual influence of operation on the security level.
>
> **Security level**: The security value for a system that is in use and, thus, have its operational aspects included in the model.
>
> **Risk level**: When the contextual environment, i.e. threats and assets, are considered, an assessment will yield a risk level.

Thus, the problem of assessing system securability is focused on the security mechanisms implemented in systems. It does not include the operational aspects yielding the actual security levels of systems or the context and information characteristics required to decide the security risks.

The securability is a characteristic of the design of an information system, including technical, organizational, and individual aspects, aiming at an estimate of the level to which systems can be secured during operation and the effort required to achieve a certain level of security. Thus, the securability is constant as long as the design is not changed.

The security level is a characteristic of an implemented information system in operation, including technical, organizational, and individual aspects. During system operation, the current security posture is the major concern, e.g. which nodes are functioning and trustworthy, which users are active and how are they connected and authenticated. Thus, the security level of a particular system can change whenever the implementation is altered, that is with system reconfiguration, or with other events affecting the security posture of the system. For example, when a new user account is added, the security level is affected, although with sensible access control mechanisms, the influence may be insignificant. If a user account is locked because the (legitimate) user fails to provide the correct password, the security level is affected (hopefully mainly considering availability). Since individual and organizational aspects are considered to be part of the information systems, the security level is still independent of the context of systems.

If the operated system is put in a context, risk levels can be estimated. Apart from security level, risk depends on antagonists and the possible damage caused by security breaches. Thus, information value has to be considered. Efficient evaluation of risk levels enables powerful risk management to be performed.

Securability, security level, and risk level are measured on different scales and, thus, are not possible to compare. Naturally, the measures can be compared using some reference transformation, but then the operation and environment have been modeled in some way. A metaphor is car, driver, and road. A car has a performance, which can be estimated in various ways, for example, the torque of the engine can be measured to give an estimate of the power of the car and its fuel consumption. However, in reality, the performance of the car is highly dependent on the driver and the environment (road).

Two more terms, related to security assessment, that are used in the report are:

Security value: Security value is the common term used to denote securability, security level, or risk level.

Security strength: Security strength denotes the ability of a system to uphold the security standards specified for the system or, even more vaguely, just a relative term indicating the general standard of security enforcement in the system.

2.2 Security Metrics

It is essential to be able to define what is actually meant when security is assessed. Security metrics is a term that can be, and has been, used for this purpose. The need for metrics is emphasized by researchers (ACSA, 2002) as well as industry (Secmet, 2004)

Definitions of Security Metric

In literature, several different definitions of security metric can be found.

Occasionally, metrics are considered to be synonymous with a measure or a sequence of measures (Leung, 2001), although security metrics by the nature of security are likely to be subjective rather than objective (Vaughn et al, 2003). Along those lines, Alger (2001) states that "the relationship between measures and metrics is twofold: metrics derives from the analysis of measures, and metrics contributes to the making of meaningful decisions and the identification of meaningful conclusions." Payne (2001) captures this in the statement "measurements are objective raw data and metrics are either objective or subjective human interpretations of those data."

According to Swanson et al (2003) "Metrics are tools designed to facilitate decision making and improve performance and accountability through collection, analysis, and reporting of relevant performance-related data." Thus, metrics are considered to be something more than pure measurement. Concerning security Swanson et al (2003) state that "IT security metrics must be based on IT security performance goals and objectives. […] IT security metrics monitor the accomplishment of the goals and objectives by quantifying implementation of the security controls and the effectiveness and efficiency of the controls, analyzing the adequacy of security activities, and identifying possible improvement actions." This is much more related to the use of the term security metric in this report.

The general conclusions presented in the proceedings of Applied Computer Security Associates (ACSA) Workshop on Information Security System Scoring and Ranking (ACSA, 2002) reveals some aspects of the nature of security metrics, although the term IS* was used instead of security metric. Some important observations are:

> "No single IS* will successfully quantify the assurance present in a system. Multiple measures will most certainly be applied and they will need to be refreshed frequently." (ACSA, 2002)

This follows from the fact that the concept of security is context dependent. For example, a system that is considered to have an adequate level of security in one environment may be considered inadequate in another context.

> Quality of software, architectures and designs chosen, tools used to build systems, and requirements specified are of great importance.

The system development process is of great importance for the securability of the resulting system. Thus, considering security assessment it is important not only to consider security during system development, which requires novel approaches to system development efficiently incorporating IT security design issues (Siponen, 2002), but also to address system development during system security assessment.

> "Processes, procedures, tools and people all interact to produce assurance in systems. IS*s that incorporate these aspects will remain critical to successful IT system operation." (ACSA, 2002)

Considering less than all aspects of systems will only generate partial solutions to the security metrics definition and security assessment problems. However, such partial solutions may result in considerable contributions to the overall goal of security assessment.

In this report, the term security metric is used with the following meaning:

> **Security metric**: A security metric contains three main parts: a magnitude, a scale and an interpretation. The security values of systems are measured according to a specified magnitude and related to a scale. By this the correspondence to meter or foot when measuring length or distance is achieved. The interpretation prescribes the meaning of obtained security values.

To further define a security metric the following issues have to be addressed:

- **What system properties are to be measured?**
  Decide on relevant security properties to parameterize. If measuring the weather, how warm it is might be a relevant property.

- **What measurable magnitude is to be measured for each parameter?**
  Decide on a suitable magnitude for each parameterized property. Continuing the previous example, how warm it is, might be measured as the temperature in the shadow.

- **What representation is to be used for each magnitude?**
  Decide on a suitable *unit* and *scale type* for each magnitude. Continuing the example, the temperature might be represented as a ratio scale, as Kelvin, or as an interval scale, as Fahrenheit or centigrade. (Roberts, 1979)

It is, again, important to remember that securability, security, and risk, due to their complex nature, always need to be aggregated magnitudes, depending on more detailed and measurable properties (Wang and Wulf, 1997). Thus, viable system security metrics cannot be as straightforward as for example meter or time. To capture the complexity of security a large number of different system properties have to be incorporated. If these properties can be measured, a vector can be used to express security values of systems or system components. If the values of such vectors can be aggregated, higher level security values of systems can be constructed. However, these values will not have any intuitive meaning. This is problematic since the lack of intuitive meaning, at least initially, eliminates the interpretation part of metrics. Eventually, an interpretation can be developed through experience and comparison with other results.

Examples of Proposed Security Metrics

As a contrast to the view of the authors of this report and others (ACSA, 2002; Swanson et al, 2003) that metrics need to be conglomerates of measures, there are proposed metrics aspiring to be more or less universal and relatively intuitive. In this section, some examples of such metrics are mentioned.

Kovacich (1997) proposes a method that can be characterized as "Management by metrics", that is, it is more targeted towards deciding the budget of the IT department than assessing the security values of systems.

Schwartau (2001) proposes time as the metric for security. The reasoning is that protection always will be inadequate for a skilled opponent and, thus, the abilities of detection and reaction are vital.

Leach (2004) proposes an approach called threat-based security engineering, TBSE, where the metric is the probability of a successful attack. The method requires that the ability of the threat to penetrate and the ability of a countermeasure to resist are quantified.

Wood and Bouchard (2001) propose *red team work factor* as a security metric. Red team work factor is defined as "an estimate of the effort required by an adversary to achieve an adversarial goal (flag). This estimate should include all costs associated with a particular attack, including preparation time, attack time, and $ expenditure for equipment, information, access, and assistance." Unfortunately, there are many reasons why red team work factor is not a viable security metric. For example, preparation time will vary drastically depending on the experience of the red team members. Moreover, if a novel method to breach the security of a system occurs, all previous results regarding work factor will be useless.

3. Approaches to System Security Assessment

Aiming at assessing the IT security of systems, several approaches have been proposed. The characteristics of these approaches are diverse depending on several causes. Naturally, the background and expertise of the originators have a large influence as well as the targeted users, the targeted use of the results, and the estimated amount of work necessary to achieve these results.

To describe the character of approaches to system security assessment four general characteristics have been identified. These characteristics, called system observing, system testing, security functionality modeling, and system structure modeling, are described below. Approaches to system security assessment may have several of these characteristics.

The main purpose of this report is to describe an ongoing research effort aiming at a structural approach to system security assessment. Consequently, approaches with that characteristic will be further discussed in the following chapter.

3.1 System Observing

System observing is a characteristic indicating that the system to be assessed is viewed from the outside, that is, as a black box. Thus, internal characteristics of the system are not considered. For example, considering a user authentication module of a system, the frequency of false positives and false negatives are measured rather than organizational, operative, and technical merits of the design and implementation of the authentication module.

System observation is often used in so called security metrics programs (Swanson et al, 2003; Payne, 2001). For example, Payne (2001) uses “Current ratio of virus alerts to actual infections as compared to the baseline 2000 figure” as an example metric. Usually, security metrics programs include other kinds of metrics than observing.

3.2 System Testing

The two main approaches to system security assessment by testing are based on the use of vulnerability scanners and red teams. The number of vulnerabilities detected by vulnerability scanners can be used to form security metrics and assessments based on those metrics. The effort required by red teams to achieve certain goals can be used for assessments based on adversary work-factor (Schudel and Wood, 2000). However, there are many questions regarding the usefulness of security metrics based on adversary work-factor, as discussed in the chapter on Security metrics

above. Since security assessments using adversary work-factor will be based on those metrics, the same arguments can be used against such assessments.

3.3 System Security Functionality

In system assessment approaches based on system security functionality, the system is scrutinized to identify the security mechanisms and measures used to prevent security violations. Compared to system observation, the emphasis is on the mechanisms used in the system, which may include organizational, individual, operational, and technical aspects, rather than the system behavior that can be observed from the outside. For example, considering a user authentication module of a system, like in Section 3.1, the focus of the current approach is on the strength of the design and implementation of the authentication module and supporting mechanisms and processes. Thus, there may be a comparison between different schemes for user authentication, such as token-based and knowledge-based. For a specific scheme, system-specific properties may be examined, such as the presence of password quality checks in a knowledge-based approach.

Thus, the classical check-list approach falls into this category. However, a straightforward check of the presence of security functionality and correct configuration will not result in system security values but rather in statements about the functionality and configuration of the system. To produce viable security values the strength of the security functions and the quality of the configuration should be used as the basis for a metric.

Alves-Foss and Barbosa (1995) propose the use of a method called System Vulnerability Index (SVI). The method considers general system characteristics. Indeed the goal is to find system characteristics general enough to yield system independent values. Thus, a specific SVI would reveal the vulnerability of systems based on a specified set of SVI rules. The method does not call for structural modeling of systems. It heavily relies on the validity of the specified set of SVI rules, so called certainty factors corresponding to the SVI rules, and the equal importance of the SVI rules.

Wang and Wulf (1997) present a framework for estimation of scalar values on high-level security attributes. The approach targets system-wide security measurements, assuming the existence of security values for system components. A decomposition method, that can be used to derive measurable attributes from the notion of security in the corresponding context, is described. However, what attributes are measurable and how to actually measure them is not revealed. A method to calculate weights in the resulting tree is presented, together with some functional relationships that can be used to model interactions between these factors. Component sensitivity analysis is introduced as a means to find sensitive components and possible flaws in the

system model. Thus, the framework includes an approach to combine the strength of security functions into scalar security values. An important remaining issue is how to turn the scalar values into meaningful security values, that is, how to create a metric. Moreover, the lack of an explicit modeling of the system structure will make it difficult to use the method in the context of distributed information systems.

The Heimdal framework described later in this report uses the Security Functional Requirements (SFRs) part of the Common Criteria (CC, 1999) to calculate security values for system components. Although being a component, rather than system assessment method, Heimdal presents some novel ideas on how to calculate security values from specifications on required security functionality. This differentiates the use of SFRs within Heimdal from the use in CC, where it is used as a checklist to verify the presence of specified security functionality.

3.4 System Structure

The system structure characteristic of assessment approaches implies that the assessed system is considered as a set of system entities. The interaction between entities is modeled with a set of relations. Thus, the assessment is based on the security values of the entities and the structure of the system, which is captured by the relations between the entities.

We believe that system structure must be considered in order to be able to produce high quality system security assessments. Consequently, only approaches with this characteristic will suffice. This does not imply that other approaches are of no value. On the contrary, they can be valuable in revealing vulnerabilities in systems. Moreover, structure by itself will not be able to produce security values. Thus, efficient methods will have to incorporate other system assessment characteristics as well.

Clark et al (2004) propose an approach based on a "multi-stage attack modeling framework". The framework supports the modeling of vulnerabilities, network structure, and attacker capabilities to enable elaborated system vulnerability analysis. The framework appears to be powerful although it is vaguely specified in the paper. The presence of physical and logical relations between network components is mentioned although not clearly specified. "Physical and logical topologies model network communication pathways. […] Logical relationships overlay network topologies, describing special communication and trust patterns."

Oman et al (2004) proposes the use of graphs to model the security and survivability of systems for Supervisory Control And Data Acquisition (SCADA). The approach targets threats based on attacks against these systems, rather than the whole IT

security posture of systems. Still, it provides an illustration of the potential of structural approaches to system security assessment.

Although more targeted at design, the approach of Blobel and Roger-France (2001) is interesting. Blobel and Roger-France (2001) propose the use of conceptual layered and domain models during the analysis and design of secure health information systems. The conceptual and layered model considers concepts, services, mechanisms, and algorithms. The concepts, which can be handled separately, are communication and application security. These concepts are realized using security services protecting against specified security threats and risks. When the necessary security services have been identified, use cases are used to find the set of services needed for specific cases and select the security mechanisms required to implement these services. Domains are used to cluster system components with related characteristics, e.g., regarding organization, location, or technical properties.

4. System Security Assessment Framework

Structural methods are based on the concept of systems consisting of interconnected entities. A structural system security assessment method combines the security values of the entities according to their specified relations in order to assess the security posture of the system. Thus, the system security value should reflect both the security values of individual system entities and the structure of the system.

Based on the structural approach, a framework for system security assessment is introduced in this chapter. The purpose of the framework is not to provide a final solution for calculations of system security values but to work as a reference for methods aspiring to assess the security posture of systems. In Section 4.4, the framework is used to derive a method for assessment of information systems and to assess the limitations of the method.

4.1 Terminology

In this section, a number of terms relating to system security assessment are defined.

Security assessment is the process of deciding the security values of systems, system entities, and system processes. A framework for the security assessment of system entities (components) is described in Part 2 of this report. System security assessment[1] is the topic of this part of the report and can be performed in several different ways, as described in Chapter 3.

Security metrics are used to describe the meaning of security and enable security measurements and analysis to produce security values. As defined in Section 2.2, security metrics consist of both a scale and an interpretation of values on the scale.

Security posture is the actual state of a system, entity, or process regarding security, that is, what the system security assessment aims to describe, using security values and metrics.

Security value is the collective term for securability, security level, and risk level, described in Chapter 2. Security values can be associated with systems and their entities. System security values are the target of the system security assessment.

[1] In this report, the terms assessment and evaluation are used synonymously, although the aim has been to use assessment throughout the text to be as unambiguous as possible.

System entities are used to describe subjects, objects, or subsystems that perform tasks in a system and the tasks themselves. Examples of system entities are organizational units, users, computers, smart cards, and authentication processes. Entities are divided into constituents and processes, see Figure 3 below. The fact that the concept of system entities includes processes may differentiate the use of the term from other texts. The complexity of the entities used to describe a system decides the abstraction level of the system model.

System constituents are used to describe parts of which a system is consisting. Constituents are not processes, but may be acting in processes. Most security assessment efforts this far mainly considers technical constituents, but the proposed framework for system security assessment acknowledges the need to consider more general constituents; that is, constituents may be for example organizational units and individuals.

System elements are system constituents that can be assessed without the need of further partitioning.

System processes are used to describe activities in a system performed by system constituents. Processes can be divided into elementary processes. Elementary processes refer to the processes that are not further divided in the system model. They are not necessarily indivisible but there are methods enabling their assessment without the need of further partitioning.

System structure results from the relations among system entities.

System relations describe security-relevant interactions and dependencies between system entities.

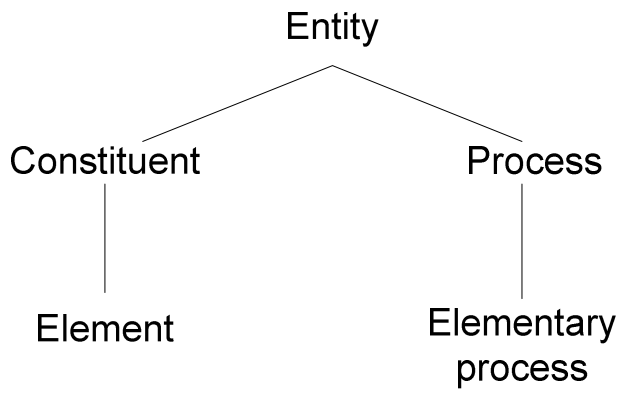


Figure 3: Relations between entity and subordinate terms.

4.2 Hypothesis

The framework introduced in this chapter is based on the following hypothesis.

With the knowledge of:
1. the security values of all security-relevant system entities and
2. all security-relevant relations between system entities,
security values for the corresponding system may be decided.

In other words, the assumption is that an ability to decide the security values of different parts of a system and to decide the relations between these system parts affecting the security will result in the ability to calculate overall system security values. Hence, the importance of the entities and processes realizing the system are acknowledged together with the influence of the relations between these entities and processes. That is, two systems with exactly the same entities and processes, but different structure (relations) may have different overall security values. Reversely, two systems with the same structure, but different entities and processes in that structure may have different overall security values.

4.3 Framework

The purpose of the framework is to describe how a structural approach to system security assessment is achieved. Thus, it will not thoroughly specify the type of system entities etc. to be used, but discuss the different possibilities. Ideally, the framework can be used as a guide when a structural approach is taken and a tool to classify different methods. The general workflow of the framework involves two main tasks:

1. model the system using predefined entities and relations and
2. use the system model to assess the security of the system.

Since the framework is general, several parameters have to be set in order to transform the framework into a method. These parameters relate to:

- system scope,
- the sets of entities and relations available to model the system, that is, the modeling technique to be used,
- methods assessing the security values of elements and elementary processes, and
- methods aggregating the security values of entities into the desired overall system security values (this implies the establishment of adequate security metrics).

The framework is illustrated in Figure 4. Each part of the framework is discussed in the following sections.

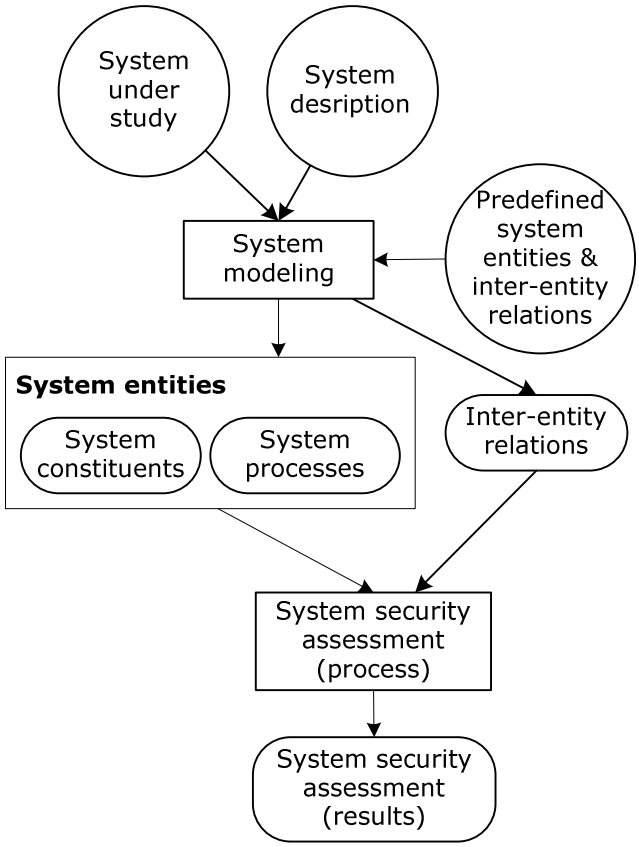


Figure 4: A framework for system security assessment.

Input to System Modeling Process

To decide the scope of the system under study is a non-trivial task. Firstly, the concept of system is not generally well-specified, for example, does it include technical, organizational, or human aspects? Secondly, the interfaces to other systems have to be specified. Thirdly, the system to be assessed can be a system in operation or a future system currently being developed. This affects both how the data required to model the system can be acquired and the purpose of the system security assessment. Fourthly, systems are, or will be, continuously developed demanding recurring reviews of the documentation of systems. The documentation available is referred to as the system description. Fifthly, all system descriptions are themselves models, more or less accurate, of the system. These models are, often, not only incomplete but even contradictory. For example, the perception of a system varies between individuals.

All the above issues have to be handled to enable efficient system modeling. Possible sources of information are, for example, organizational charts, process models, interviews, and technical documentation.

The set of predefined system entities and relations decide at what levels of abstraction the system can be modeled. The set can be viewed as a set of templates that are used to instantiate the entities and relations of system models. In some cases, or system security assessment approaches, this set may consist only of concepts originally described by the authors of the method. In other approaches, the predefined system entities and relations may be models themselves allowing semi-automatic instantiation of structures of entities and relations.

System Modeling

Although there is a common objective to model the security-relevant characteristics of systems, the model may be at different levels of abstraction. An extreme approach, in this regard, is to view the system as a single entity (possibly interacting with other entities outside the assessed context). This corresponds to the non-structural approaches discussed in Chapter 3, since the structure of the system itself is neglected. In more detailed (structural) approaches, the system is modeled as several entities and their security-relevant relations.

The sets of entities and relations constitute the system model and are the basis for the system security assessment. In some cases, the entities of a system are complex themselves and, possibly, referred to as subsystems, which in turn are divided into entities. The entities are divided into constituents and processes. The constituents consist of elements and processes of elementary processes.

Correspondingly, the system could be considered a system of systems. However, although complex, the system may always be considered a part in a higher-level system. Thus, there is a hierarchy with several levels of abstraction where an assessed system can be modeled. Possibly, there is more than one level of systems, entities, and, finally, elements and elementary processes. The concept of different levels of abstraction in a system model is illustrated in Figure 5.

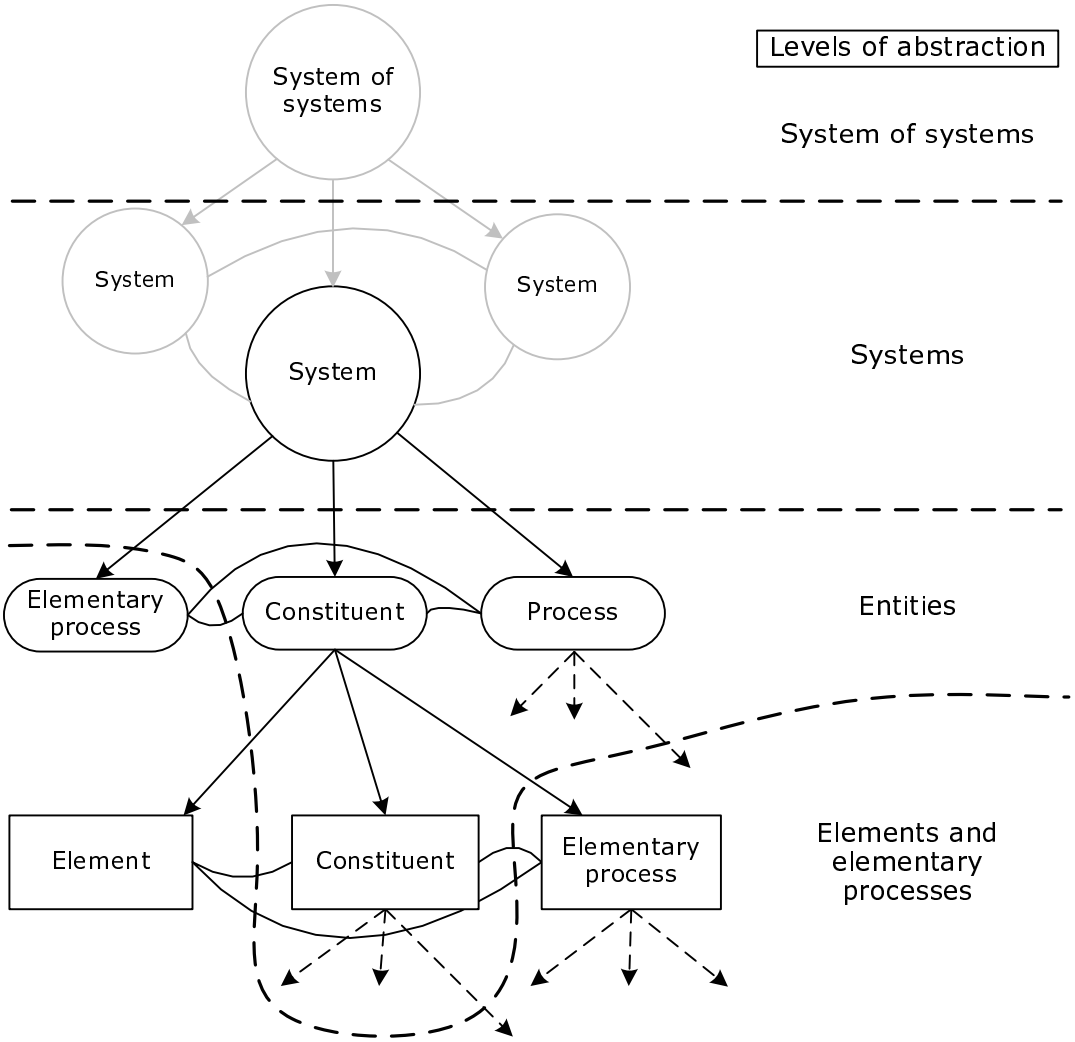


Figure 5: Abstraction levels in a system model.

As an example of possible levels of abstraction in a system model, consider Figure 6, from (Peterson, 2004), which specifies three levels of abstraction in a system. In principle, level C may consist of several levels of entities[2], that is, level D and so on can be added. In (Peterson, 2004) the term component is used in the meaning technical entities. Consequently, humans, organizational units, processes, etc. are not considered so far in this approach.

[2] The approach of Peterson, which is further described in Section 4.4 and Appendix B, does not include the concept of processes.

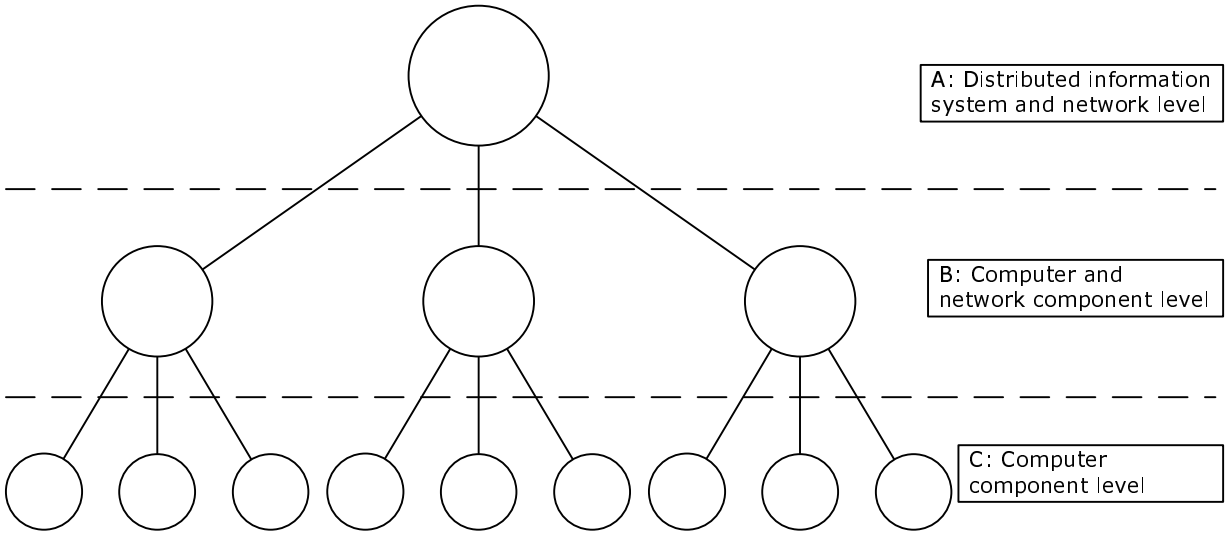


Figure 6: Levels of abstraction referred to by Peterson (2004).

To actually enable modeling of systems using a structural approach, where system elements can be modeled hierarchically, the set of possible system entities and relations has to be specified, that is, a modeling technique has to be established.

System Entities and Relations

The way system entities and relations are modeled has a large influence on the amount of information that can be captured by system models, the amount of work required to model systems, and the comprehensiveness of the models. A more detailed model is not necessarily more adequate, depending on the precision of the security-related data captured by the model and the capability to aggregate these data into meaningful system security values.

Several different approaches use entities, although they are usually called something else, to model system with the aim of security assessment, or at least analysis. Examples of different levels of entity specifications are:

- Computers and network components (Hunstad and Hallberg, 2002b), (Clark et al, 2004)
- Supervisory Control And Data Acquisition (SCADA) system devices (Oman et al, 2004)
- Access interfaces, such as, SCADA user interface, local terminal, and remote access, (Oman et al, 2004)
- Networks and network links, such as, Internet, WAN, LAN, Ethernet, telephone line, and proprietary wiring, (Oman et al, 2004), (Hunstad and Hallberg, 2002b)

- Security services, such as, firewalls and intrusion detection systems (IDSs), (Blobel and Roger-France, 2001), (Eschelbeck, 2000), (Whitmore, 2001)

Processes can be used to reveal the importance of the security of their enabling entities. Moreover, processes reveal security relations between entities. For example, if the security of a smart card used to implement an authentication process is violated, the security of terminals relying on the smart card is jeopardized. Figure 7, from (Hallberg, 2000), illustrates how a model of the smart card-based authentication process can be used to identify entities central to the security of the corresponding system.

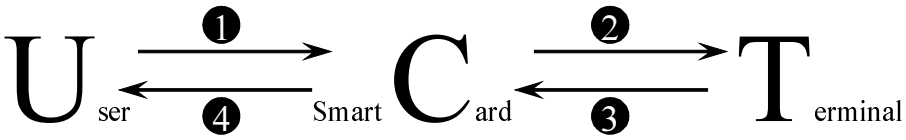


Figure 7: The traditionally identified entities involved in a smart card-based authentication process, that is, the user, smart card, and terminal. The numbered edges represent the authentication steps used to establish a chain of trust between these entities.

There are a large number of possible processes within systems. As stated in 4.1, processes describe activities in a system performed by system entities. Processes are normally designed as part of the system, such as for example

- authorization,
- handling of protocols, and
- PKI-handling processes.

There are system-related processes which normally are not designed to be part of the system, but may affect it although. Attack processes are an example of this. Such a process has to be handled by a set of different processes and entities designed into the system.

Relations are used to describe security-relevant interdependencies between system entities. Examples of different possible relations in a system are:

- SCADA system devices access paths (Oman et al, 2004),
- physical connections, that is, the ability to communicate directly, and
- logical connections, that is, dependencies or indirect communication paths.

Using standard entities, which have already separately been assessed, is expected to simplify the process of system security assessments. Likewise, defining and using standard relations is also expected to enhance and simplify system security assessments. By adjusting parameters of standard entities and standard relations, individual or close to individual characteristics may be modelled faster and more efficiently.

System Security Assessment Results

The framework for system security assessment is expected to produce an aggregated result which for the actual studied system and environment in an appropriate way describes the security level of the system. Aggregation does not necessarily imply that the result should only be a single value. For example, it may be a system map specifying the weak spots of the system.

There is a need to consider what kinds of results are needed in different situations. For example, what the appropriate metrics are will probably be guided both by what in the best way describes the entities and relations of the system, but even factors regarding the environment of the system and of which categories of users are interested in the results.

System Security Assessment Procedure

To produce the system security assessment results, a procedure for system security assessment is necessary. Security relevant information regarding all entities of a system and their inter-entity relations are input to the procedure of system security assessment. With adequate system models the assessment procedure have to specify how the model should be interpreted in order to produce the desired assessment results. However, the aggregation of results will be difficult to perform in order to achieve meaningful results. For example, if system elements are assessed using different scales, then an overall system security value calculated as an arithmetic mean will be meaningless (Roberts, 1979). The design of assessment methods producing meaningful results heavily depends on the ability to formulate system security metrics, as discussed in Section 2.2. The formulation of adequate system security metric and, consequently, methods appropriately achieving aggregated system security assessment results are open research questions. A proposal for a simplified procedure to achieve such measures is discussed in Chapter 4.4.

4.4 Example – CAESAR a Method for System Security Assessment

The CAESAR method described in (Peterson, 2004) illustrates one possible way of performing system security assessment, referred to as evaluation in the example.

Details of the method are described in Appendix A and in (Peterson, 2004). The scope of the method is limited to technical aspects of the assessed systems. Entities, which in this example are referred to as components, are modeled as either traffic generators (computers and public networks) or traffic mediators (firewalls, routers, proxies and hubs). Processes are so far not modeled in this approach. Relations are modeled as physical or logical relations.

The main goal of the algorithm is to calculate the *overall security level* (OSL) of the system, but many partial results are interesting by themselves and may be analyzed to understand how different factors affect the security level of different parts of the system.

To be able to calculate the overall security level of the system, it is necessary to perform an evaluation of each component of the traffic generator class. Such an evaluation results in a *system-dependent security level* (SSL) for each traffic generating component.

Figure 8 includes a general overview of which modeled and calculated properties that are necessary to perform an evaluation of a system.

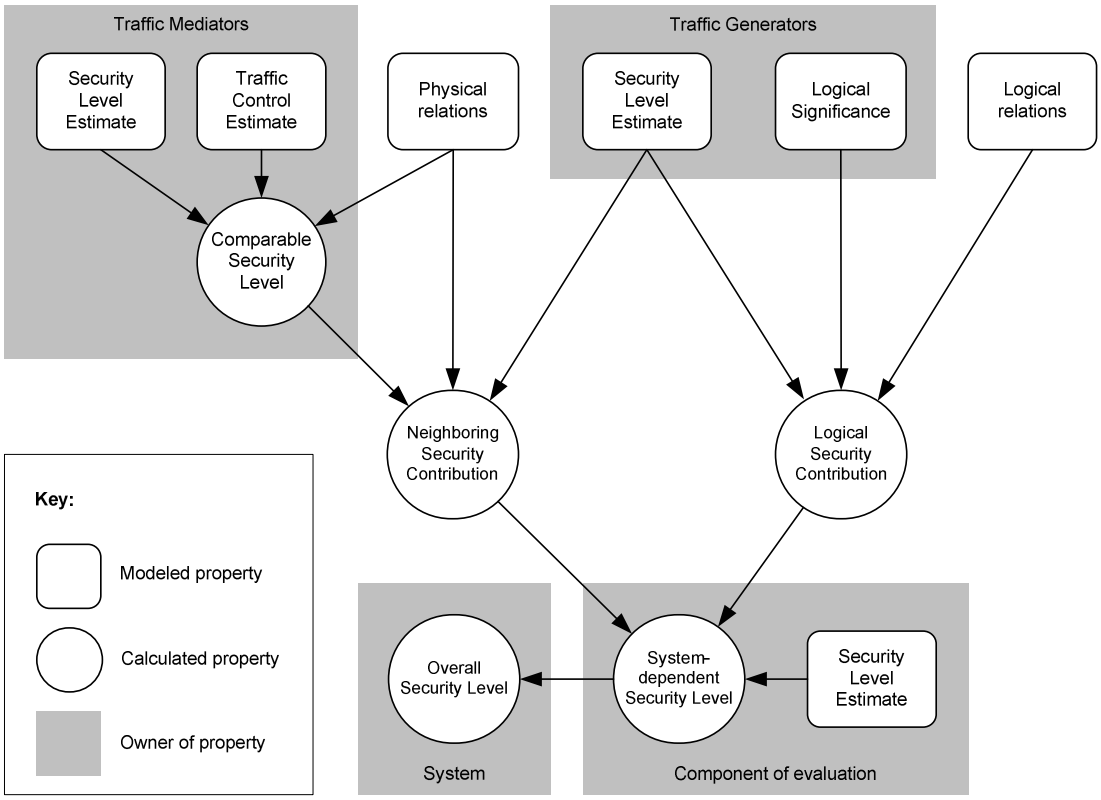


Figure 8: The CAESAR evaluation algorithm's main concepts and their relations

The system-dependent security level of each traffic generator, from here on referred to as the component of evaluation, is calculated by aggregating the neighboring security contributions of all physically related traffic generators and the logical security contributions of all logically related traffic generators acting as servers to the component of evaluation.

The neighboring security contribution of a physically related traffic generator is calculated by aggregating the security level estimate of that traffic generator, with all traffic mediators that are on the physical path or paths between the traffic generator and the component of evaluation taken into account by aggregating their comparable security level.

The logical security contribution of a logically related traffic generator acting as a server to the component of evaluation is calculated by aggregating the security level estimate of the traffic generator and the logical significance of the logical relation that connects the traffic generator with the component of evaluation.

The overall security level of the distributed system is then calculated by aggregating the system-dependent security level of all traffic generators. The algorithm may also be described on a shorter form, as expressed below:

```
For each traffic generator, Ce, in system
        for each traffic generator, Cg, physically related to Ce
                calculate the aggregated CSL between Cg and Ce.
                calculate the NSC from Cg to Ce using CSL.
        for each traffic generator Cg, logically related to Ce
                calculate the LSC from Cg to Ce.
        calculate the SSL of Ce using all CSL, all LSC, and the SLE.
Calculate the security level of the system.
```

How these modeled and calculated properties are aggregated is further described in Appendix A.

4.5 Categorization of System Security Assessment Approaches

The framework introduced in Section 4.3 can be used to categorize system security assessment approaches. In this section, CAESAR is categorized. As discussed in Section 4.3 above, the characteristics of the approach regarding system scope, modeling technique, entity assessment methods, aggregation of security values, and security metrics are used to perform the categorization. The names of the considered aspects are in bold in the following line-up:

- system scope,
  - **system aspects** considered, that is, technical, organizational, human, operational, or contextual aspects,
  - **external interfaces**, specification of interfaces to other systems,
  - **system status**, that is, is the considered systems in operation or currently being developed,
  - **system descriptions**, are specification of system description considered and/or required.
- modeling technique,
  - **system entities** considered,
  - **system relations** considered,
  - **abstraction levels**, can the system be modeled at different levels of abstraction?
  - **hierarchical models**,
- **entity assessment methods**, that is, methods used to assess the security values of elements and elementary processes,
- **security values aggregation methods**, and
- **security metrics** used to enable the interpretation of the result.

The result of categorizing CAESAR according to the criteria specified above is included in Table 1.

Table 1: Categorization of CAESAR according to the framework.

Criteria	CAESAR characteristic
System aspects	Technical
External interfaces	Only addressed through the modeling of public networks
System status	Not specified
System descriptions	Not specified
System entities	Traffic generators (computers and public networks) and traffic mediators (network components)
System relations	Physical and logical
Abstraction levels	Computer and network components
Hierarchical models	Not supported

Criteria	CAESAR characteristic
Entity assessment methods	Not specified
Security values aggregation methods	Arithmetical operations on scalar values
Security metrics	Not specified

4.6 Discussion

In this chapter, a framework is specified. The purpose of the framework is to work as a reference for and enable the categorization of methods aspiring to assess the security posture of systems.

As an example of a possible approach to system security assessment, the CAESAR method was presented. CAESAR enables the calculation of scalar system security values.

To illustrate the possibility to categorize approaches using the introduced framework, CAESAR was scrutinized. The results show that CAESAR does not include several important characteristics that are considered as important for efficient structural system security assessment.

An observation by the authors is that the most critical and challenging issue of system security assessment is to capture the security-relevant relations between system entities. For example, the consequences of physical relations are hard to quantify. A plausible approach would be to substitute physical relations with logical relations. That is, a physical relation would result in the instantiation of a set of logical relations capturing the security-relevant consequences of a physical relation. Properties of the physical relation and the entities connected by it decide which logical relations to be included in the set. As a consequence, physical relations could be viewed as approximations of the security influence resulting from sets of logical relations.

5. An Approach to Component Security Evaluation

A method for evaluation of the security of components in distributed information systems was introduced by Andersson et al. (2003) and is outlined in this chapter. Limitations of and improvements to the method are discussed.

5.1 Evaluation of the Security of Components in Distributed Information Systems

The method utilizes the Security Functional Requirements (SFRs) of Common Criteria[3] (CC, 1999) for the estimation of security values for components. This is accomplished through (1) the identification of the SFRs relevant to the security of the component and (2) the assignment of values to the identified SFRs characterizing their strengths. The assigned values can be mapped to the confidentiality, integrity, and availability characteristics (CIA) or the prevent, detect, and react (PDR), abilities of the component resulting in more meaningful security values.

Since Andersson et al (2003) changes the purpose of the SFRs of CC, from description to evaluation, some changes in the structure of the SFRs have been made. The most significant alteration is how to regard the ordering of the lowest level CC SFRs due to the overlapping of some of them; some requirements are merged, others split. The process of merging and splitting is detailed in (Andersson et al, 2003).

Mapping component characteristics to CC SFRs

To determine which SFRs are relevant for the security level of a specific component, or class of components, CC Protection Profiles and other reliable information are used. The SFRs assumed to be irrelevant are assigned a NULL value. Once the relevant SFRs have been determined, estimated security values are assigned to the set of security functions that are actually included in the component. Those not included are assigned a zero value.

Security Evaluation of CC SFRs

When the security functions have been assigned their values, there are a few different ways in which they can be presented. For each of the 11 SFR classes in CC, one may choose to present any of the following:

[3] Relevant abbreviations and other details regarding Common Criteria are presented in Appendix C.

SFRs table presentation

One possibility is to present the result of the evaluation as the resulting SFRs table. This leaves an experienced evaluator with a detailed picture of the securability of the Target Of Evaluation (TOE). On the other hand, it may seem somewhat complicated and thus unclear to less experienced people.

CIA or PDR vector presentation

Another solution is to translate the values into a more accepted and recognizable terminology, such as CIA and PDR. This requires a mapping between the SFRs and their CIA or PDR properties. For further details, the reader is referred to (Andersson et al, 2003).

Single index representation

A third solution is to traverse the values for the SFRs upward, yielding results at more general levels, and finally reach a measurable security value at the top of each of the 11 classes of SFRs. For further details, the reader is referred to (Andersson et al, 2003).

Calculating security values

Regardless of which of the representations explained above is chosen, calculating a security value for a system component can be done by following the steps below:

1. Choose which of the above explained ways to represent the security values.
2. Calculate mean values of the above chosen types for every family.
3. If there are SFR components that should be prioritized before others, their security values should be multiplied with a weighing matrix to reflect this prioritization.
4. NULL-values do not affect the calculations and should simply be ignored.
5. Calculate mean values for every class, or corresponding concepts depending on the chosen representation.

5.2 Applying the method on Windows 2000

To get a practical sense of the accuracy and relevancy of the method, this section describes its application on the Windows 2000 Professional operating system with Service Pack 3 and Q326886 Hotfix installed. In (Science Applications International Corporation, 2002), (NIST, 1999), and (NSA, 2001), the Security Target and relevant Protection Profile are found respectively. Based on these, the set of relevant security functions was established, and each function was assigned a value. In the calculations, only the CIA aspects are considered, but a PDR-mapping is also

possible. The mapping from security functions to CIA below is done according to (Andersson et al, 2003). Furthermore, no weighting matrix is used.

Table 2: Estimated security value for the class FAU (Security Audit).

ID	P_1P_2T	CIA	SV	C	I	A
FAU			0,34	0,32	0,33	0,30
FAU_ARP	P_2	CIA	0,00	0,00	0,00	0,00
FAU_GEN	P_1P_2T	CIA	0,85	0,85	0,85	0,85
FAU_GEN.1	P_1P_2T	CIA	0,80	0,80	0,80	0,80
FAU_GEN.2	P_1P_2T	CIA	0,90	0,90	0,90	0,90
FAU_SAA	P_2	CIA	0,00	0,00	0,00	0,00
FAU_SAA.1	P_2	CIA	0,00	0,00	0,00	0,00
FAU_SAA.2	-	CIA	NULL	NULL	NULL	NULL
FAU_SAA.3*	-	CIA	NULL	NULL	NULL	NULL
FAU_SAR	P_1P_2T	CIA	0,77	0,77	0,70	0,70
FAU_SAR.1	P_1P_2T	CIA	0,70	0,70	0,70	0,70
FAU_SAR.2	P_1P_2T	C	0,90	0,90	-	-
FAU_SAR.3	P_1P_2T	CIA	0,70	0,70	0,70	0,70
FAU_SEL	P_1P_2	CIA	0,00	0,00	0,00	0,00
FAU_STG	P_1P_2T	IA	0,40	-	0,6	0,40
FAU_STG.1	P_1P_2T	IA	0,60	-	0,60	0,60
FAU_STG(2)	P_1	A	0,00	-	-	0,00
FAU_STG.3*	P_1P_2T	A	0,60	-	-	0,60

Table 2 represents the Security Audit (FAU) class of the SFRs. The values assigned to the functions are based on comparisons between requirements from the Protection Profile and the stated functionality in the Security Target.

The P_1P_2T column indicates the presence of the security function in the two Protection Profiles and the Security Target. The CIA column indicates the mapping from the security function to CIA properties. The SV column shows the estimated total Security Value, and the C, I, and A columns show the estimated Security Value for the three CIA categories respectively.

If the security function is present in any of the Protection Profiles but not in the ST, it means that it is not implemented in the product, giving it the value 0. In the event that the security function is not present in any the Protection Profiles, it will be given the security value NULL and is as such not considered in the calculations – regardless of whether it is present in the Security Target or not. These security functions are indicated with grey text.

The rows with grey background represent families in SFRs, and the rows with white background represent elements in the families. The dashes (-) in the C, I and A columns indicate that the security values are inapplicable for the given category. The

security functions whose IDs are marked with an asterix (*) have been altered from the original CC functions by merging and splitting (as mentioned above).

Repeating the same calculations as above for all 11 classes of CC results in the values presented below. Classes with only NULL values (FCO and FPR) are excluded from Figure 9. The security values, total as well as for CIA, for each class are summarized in Table 3.

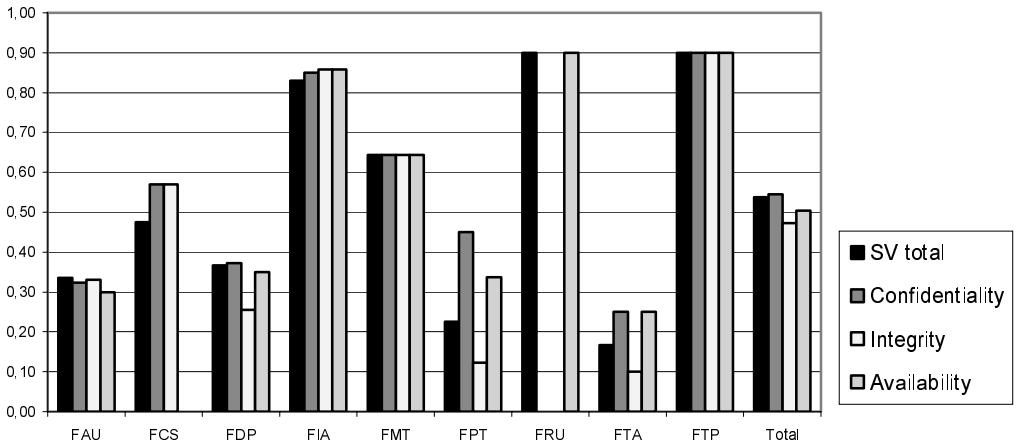


Figure 9: Security values for the SFR classes. Classes with a security value of NULL are not included.

Table 3: Security values for the 11 classes in CC.

ID	Descriptive Name	SV	C	I	A
FAU	Security audit	0,34	0,32	0,33	0,30
FCO	Communication	NULL	-	NULL	-
FCS	Cryptographic Support	0,48	0,57	0,57	0,00
FDP	User Data Protection	0,37	0,37	0,25	0,35
FIA	Identification and Authentication	0,83	0,85	0,86	0,86
FMT	Security Management	0,64	0,64	0,64	0,64
FPR	Privacy	NULL	NULL	NULL	NULL
FPT	Protection of TOE Security Functions	0,23	0,45	0,12	0,34
FRU	Resource Utilisation	0,90	-	NULL	0,90
FTA	TOE Access	0,17	0,25	0,10	0,25
FTP	Trusted Path/Channels	0,90	0,90	0,90	0,90

Interpreting the results

These results may provide indications on which parts of the operating system are well-implemented, and which are not. Even though our input data are somewhat uncertain, the zeros resulting from the comparison between the Protection Profiles and Security Targets influence the characteristics to a rather large extent.

The results can also be used to compare different products in the same category. Using the same Protection Profile, the comparison can be accurately based on the actual numbers resulting from the evaluation.

Windows 2000 has been evaluated by the National Institute of Standards and Technology (NIST) using a Protection Profile developed by National Security Agency (NSA) along with some additional enhancements. The Protection Profile has officially obtained the EAL3 assurance level, meaning it is designed for a generalized environment with a moderate level of risk to assets. Generally, it can be said that the higher the EAL of the Protection Profile, the more reliable an evaluation based on the method applied above becomes. In other words, the EAL rating does not in itself provide any relevant information about the security of the TOE, but it does correlate with the certainty of the evaluation.

5.3 Discussion

By using CC, the described method takes advantage of a systematic methodology in establishing the low-level security functionality embedded in a given product. CC is a widely spread and accepted evaluation standard.

The fine granularity of the security functions of CC traversed upwards towards the eleven classes provides a good picture of the security properties of the product. It enables comparison between different products within the same product category.

The strong connection to the Protection Profile makes the possible comparisons between evaluated products, based on different Protection Profiles, irrelevant. Because of this strong connection, the evaluation may also lack important security properties that are not included in the Protection Profile.

Even when comparing products with the same Protection Profile, the resulting values that are compared only state whether one product is better than the other; it does not state whether any of the products are secure enough, given their intended use and their environments. The method lacks the modularity needed to easily add or remove requirements and other aspects, such as users and threats. This also limits the possibilities to store parts of previous evaluations for later use.

A disadvantage with the method described above is its use of estimated security values for the security functions.

5.4 Improvements to the Existing Method

Discreet notation

The use of security values in the range from 0 to 1 may increase the theoretical precision, but will – due to the lack of means to determine these values – in practice result in a false sense of accuracy. Much work can be put into estimating these proposed values without necessarily reaching a more accurate end-result.

Figure 10 shows different estimated Security Values for Windows 2000 – extremely low, medium, as well as extremely high estimated values. They are presented only to indicate the security characteristics of differently estimated values; how they vary – and, more importantly – how they do not. The low values are derived from random Security Values on CC component level in the range [0.14, 0.29], the medium values from values in the range [0.40, 0.95] and the high values from values in the range [0.70, 1.00].

Figure 11 shows the same classes as Figure 10, but with CC component Security Values being either 0 or 1.

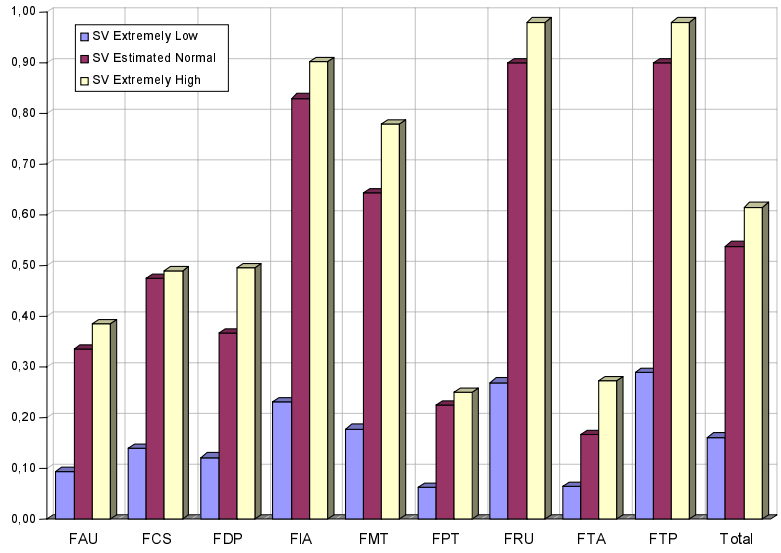


Figure 10: Security characteristics for Windows 2000 Professional with estimated Security Values ranging from 0 to 1.

The security characteristics of a product are largely formed by the Security Values that are not included in the Security Target rather than the estimated values of those that are. By using a discreet notation, where the values may be either 0 or 1, the characteristics of a component will not differ much from the continuous notation, as can be seen comparing Figure 10 and Figure 11.

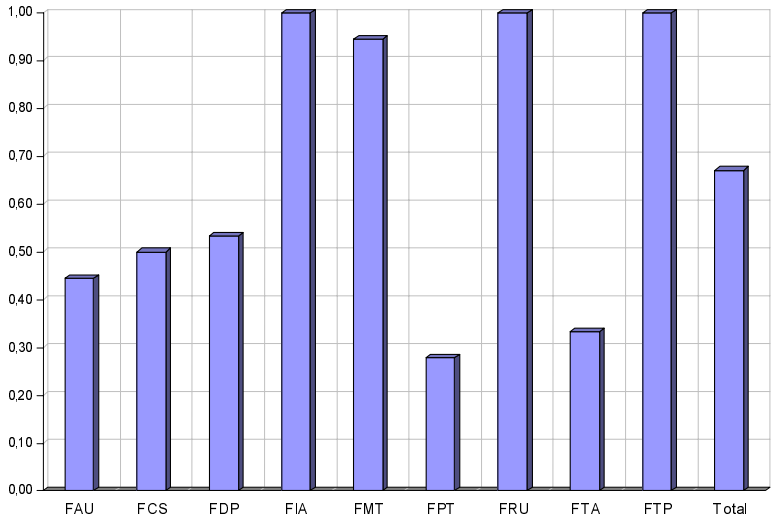


Figure 11: Security characteristics for Windows 2000 Professional with Security Values of either 0 or 1.

This may seem like a strong generalization and simplification, since a slight indication of the presence of a SFR in the Security Target will suffice to give the SFR the value 1; the discreet notation will not reflect the strength of implementation for the given SFR. However, the most significant aspects of the security of the product is identified by looking at what is not implemented, rather than looking at how good an implementation is. It is assumed that most estimated security values are closer to 1 than 0, if implemented at all.

No NULL values

As stated above, NULL values are assigned to SFRs that are not included in the Protection Profile. The reason for this is that they are assumed to be irrelevant to the TOE's security functionality. As a result, this ties an evaluation tightly to the Protection Profile, making comparison between two similar products with different Protection Profiles virtually pointless.

The SFRs that are assigned the NULL value may very well be implemented, although there are no requirements for them. If implemented, they should be assigned the value 1. The fact that some SFRs are not implemented and some are provides information about the product in a wider sense. As the product lacks or provides security in that area, the previously ignored requirements should be assigned the value 0 or 1, indicating whether they are implemented or not – thus adding information about the security functionality to the evaluation.

This way of looking at non-implemented SFRs will lower the average security value of the TOE to a seemingly unnecessarily low level given its requirements. Because of

this, a way to weight the values based on the TOE's product category is desired. A solution to this is suggested in Chapter 6.

Complete set of Security Functional Requirements

As mentioned in section 5.1, the method presented by Andersson (2003) suggests that some SFRs should be merged and others split, depending on whether some of their functionalities are overlapping. When two or more SFRs are of the same type, they are combined into a new single SFR.

If one of the requirements included in the merged requirement is implemented and the others are not, the new combined requirement is assigned a lower Security Value than if all included requirements were implemented.

This poses a problem with the discreet notation, as it provides no way of assigning a value between 0 and 1. The problem is solved by using the complete set of SFRs from CC and keeping track of the hierarchical dependencies (subsets); if a more specific SFR is implemented it should be assigned the value 1 and the other SFRs the value 0. If the more general SFR is implemented they should all be assigned a 1.

Table 4 below lists all SFRs with hierarchical dependencies (subsets). The right-most column contains the SFRs that are subsets of the given SFR.

Table 4: List of hierarchical dependencies (subsets).

ID	Descriptive Name	Hierarchical dependencies
FAU	Security audit	
FAU_SAA.4*	Complex attack heuristics	FAU_SAA.3
FAU_STG.4*	Prevention of audit data loss	FAU_STG.3
FCO	Communication	
FCO_NRO.2*	Enforced proof of origin	FAU_NRO.1
FCO_NRR.2*	Enforced proof of receipt	FCO_NRR.1
FDP	User data protection	
FDP_ACC.2*	Complete access control	FDP_ACC.1
FDP_IFC.2*	Complete information flow control	FDP_IFC.1
FDP_IFF.2*	Hierarchical security attributes	FDP_IFF.1
FDP_IFF.4*	Partial elimination of illicit information flows	FDP_IFF.3
FDP_IFF.5*	No illicit information flows	FDP_IFF.3, FDP_IFF.4
FDP_ITT.4*	Attribute-based integrity monitoring	FDP_ITT.3
FDP_RIP.2*	Full residual information protection	FDP_RIP.1
FDP_ROL.2*	Advanced rollback	FDP_ROL.1
FDP_UIT.3*	Destination data exchange recovery	FDP_UIT.2
FIA	Identification and authentication	
FIA_UAU.2*	User authentication before any action	FIA_UAU.1

FIA_UID.2*	User identification before any action	FIA_UID.1
FMT	Security management	
FMT_SMR.2*	Restrictions on security roles	FMT_SMR.1
FPR	Privacy	
FPR_UNO.2*	Allocation of information impacting unobservability	FPR_UNO.1
FPT	Protection of the TOE Security Functions	
FPT_RCV.2*	Automated recovery	FPT_RCV.1
FPT_SSP.2*	Mutual trusted acknowledgement	FPT_SSP.1
FRU	Resource utilization	
FRU_FLT.2*	Limited fault tolerance	FRU_FLT.1
FRU_PRS.2*	Full priority of service	FRU_PRS.1
FRU_RSA.2*	Minimum and maximum quotas	FRU_RSA.1
FTA	TOE access	
FTA_MCS.2*	Per user attrib. limitation on multiple concurrent sessions	FTA_MCS.1

For example: in the table above, if the SFR FRU_RSA.2* is implemented, FRU_RSA.1 which is a subset of FRU_RSA.2* is implemented as well.

6. Component Security Evaluation Framework

Based on the functional approach, a framework for component security evaluation is introduced in this chapter. The framework aspires to handle security properties of a product, as well as finding a way to scale these properties based on a number of different factors, such as the requirements of a certain product type, and the environment in which the product is intended to operate.

Regarding the use of the terms entity versus component, it should be noted that component is used when discussing technical entities; that is, component is used as a more restricted term. Although the framework is designed to evaluate components, it might even be useful for certain system evaluations. The main difference between this framework and the previously described Caesar framework is, apart from the component versus system view, the functional approach of this framework versus the structural approach of Caesar.

6.1 Terminology

For the purposes of the proposed functional Heimdal Framework, some terms are introduced and defined in this section. Heimdal may use input from Common Criteria (CC) evaluations, but does not need to. Therefore, security functionality is here described using the terms Security Classes, Security Groups and Security Features, see Figure 12.

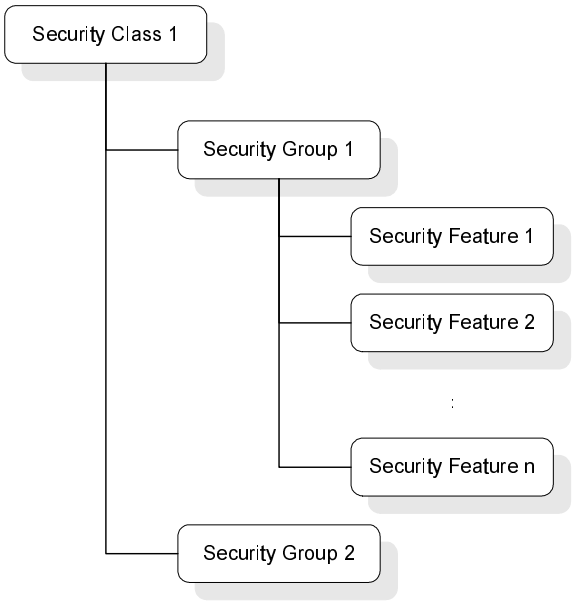


Figure 12: Decomposition diagram for a Security Class.

Security Class

The Security Classes are the highest level of the categorization that describes security functionality for a TOE or requirements for a category. It can be compared to the 11 CC SFR *classes*.

Security Group

Security Groups are ordered under the Security Classes. The term corresponds to the CC SFR *families*.

Security Feature

The Security Feature is used to describe either implemented security functionality or requirements related to a category. In some ways, it corresponds to the CC SFR *component*.

Security Value (SV)

A Security Value is a numerical value denoting the security for a given Security Feature, Group or Class. It can be presented either as the total value, or be divided into separate Security Values for CIA or PDR.

Set of Security Features (SSF)

The Set of Security Features represents the fundamental building block of the profiles. In this work, the Security Functional Requirements (SFRs) from CC will be used as the Set of Security Features.

6.2 Overview

The framework Heimdal is a process of evaluating a product, taking advantage of several different profiles, which are combined in steps through different operations, see Figure 13.

The security properties of a specific product, or TOE, will be summarized in a TOE Profile (TP). Combining requirements (for example from various Protection Profiles) for a given product category will result in a TOE Category Profile (TCP). Based on the TCP and an environment profile (EP), which contains user and threat properties, a reference profile (RP) can be derived. The final step in the evaluation of a TOE is to weigh the TOE profile against the reference profile, creating an evaluated TOE profile (ETP).

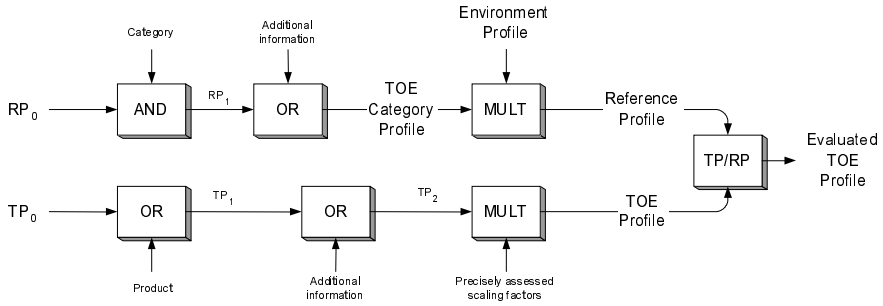


Figure 13: Overview of the framework.

The strong ties between the Protection Profiles and Security Targets in the Andersson (2003) evaluation method create a problem when it comes to comparing products in the same category but with different sets of requirements. This is solved in the new framework by separating the product's properties from its requirements, creating a TOE Profile for the former and a TOE Category Profile for the latter.

This separation results in a new modular profile-based framework. An advantage of this is that more aspects can be added. In particular, the environment of operation for the product can be taken into account.

6.3 TOE Profile

The first step in an evaluation of a product is to identify its security functionality before any requirement or environment considerations are taken into account. In the Heimdal Framework, the security properties of a product (TOE) are summarized in a TOE profile (TP). The process of developing a TOE Profile is described in Figure 14. The profile is essentially a set of values between 0 and 1, indicating which security features are implemented in the TOE.

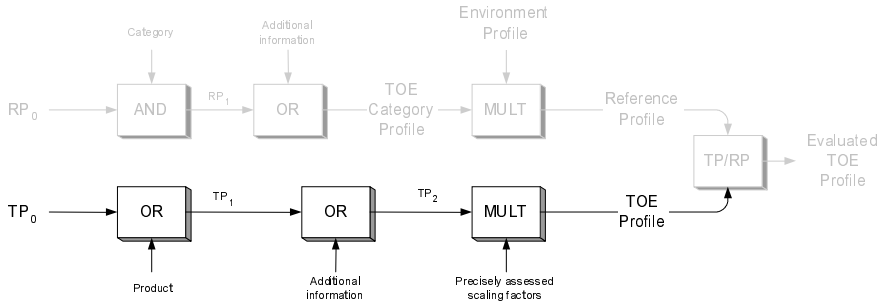


Figure 14: The process of developing a TOE Profile.

To create a TOE profile, information about the security properties of the product is required. This information can be retrieved from a CC Security Target and/or other reliable sources.

In the cases where knowledge about the strength of the implementation is good, i.e. through measurements, the security features can be assigned a value between 0 and 1, similar to the method in section 5.1.

6.4 Reference Profile

The Reference Profile (RP) can be seen as a filter, through which the TP is filtered, and thereby weighted according to important requirements. The absolute values from the TOE Profile do not state whether the product is secure or not, given its intended field of use, product category, and/or the environment of operation. An overview of the process of developing a Reference Profile is presented in Figure 15.

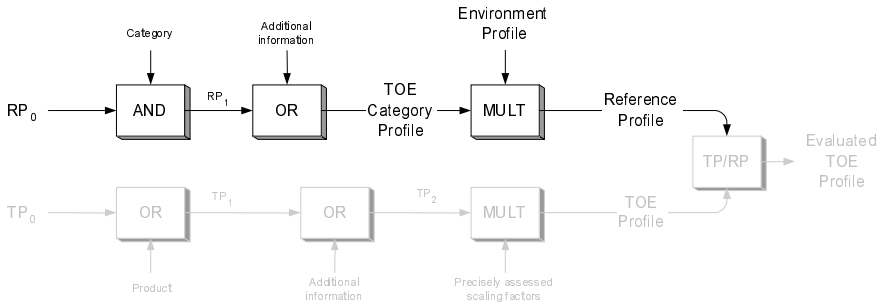


Figure 15: The process of developing a Reference Profile.

The Reference Profile can contain any information relevant to the security of the product; this includes special requirements for a certain type of products as well as external aspects, such as users and threats, as described and handled by a Environment Profile.

The requirements related to a certain category of products are summarized in a TOE Category Profile, TCP. Based on the TCP, a number of different products within the same category can be compared to each other, weighed with regards to the most important security aspects of that category. The TCP is explained in further detail in section 6.5.

6.5 TOE Category Profile

The TOE category profile is the basis for the estimation of the category's security requirements. It may include a combination of relevant Protection Profiles, as well as additional information – which gives the reference profile designer the ability to include additional requirements into the profile. The process of developing a TOE Category Profile can be seen in Figure 16.

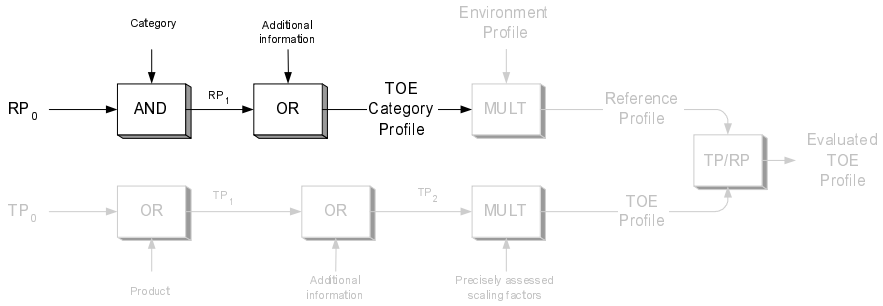


Figure 16: The process of developing a TOE Category Profile.

The process of creating a TOE category profile (TCP) is similar to that of creating a TOE profile. It takes advantage of official documents related to the CC. Information about the security requirements of a specific product category can be retrieved from a Protection Profile and/or other reliable sources. This may include the evaluator's own set of security requirements.

6.6 Environment Profile

The Environment Profile (EP) is the result of evaluating the environment in which the product to be evaluated operates in. The security environment includes the threats to security of assets that are, or are held to be, present in the environment. It also includes, but is not limited to, laws, organizational security policies, user customs, user expertise and knowledge that are determined to be relevant. It thus defines the context in which the TOE:s are used or are intended to be used.

6.7 Evaluated TOE Profile

The Evaluated TOE Profile is calculated in the following way: Each value on the Security Feature level in the TOE Profile is divided by each corresponding value in the Reference Profile. The Security Values on Group level are calculated as the mean values of their Security Feature Security Values, and the Security Values on Class level are calculated as the mean values of their Security Group Security Values.

7. Conclusions

System security assessment is a complicated issue. Security metrics matching the contextual-dependent meaning of IT security have to be specified. Systems have to be modeled in such a way that security-relevant characteristics to be measured are captured. This requires the specification of security-relevant system characteristics and adequate system modeling techniques. The structure of the system has to be captured and included in the assessment.

In this report, four general characteristics of system security assessment methods have been identified. These are system observing, testing, security functionality describing, and system structure describing. We argue that system structure has to be considered for efficient security assessments of large information systems.

To capture issues that need to be considered by structural system security assessment methods, a framework is introduced. The framework can be used to guide the design of system security assessment methods as well as to categorize existing methods.

The framework has been used to categorize the CAESAR method for system security assessment. The result shows that there is much work, research and development, to be done for the method to be able to yield efficient assessment results.

It is probably not to bold to transfer the result of the categorization of the CAESAR method to the area in general. This results in a number of topics for future work, some of those are listed below.

- The framework itself can be developed to address additional issues of system security assessment.
- The framework can be applied to other existing methods to categorize current approaches.
- To structure the process of developing system security assessment methods, system development methodology can be applied. Thus, the needs of system security assessment should be analyzed in order to facilitate the establishment of requirements and a requirements engineering process to decide the desired characteristics of system security assessment methods. The requirements engineering needs to be a continuous process.
- Methods to produce context-sensitive security metrics are needed.
- Existing modeling techniques need to be developed together with tools supporting the system modeling process.

Bibliography

ACSA (2002), *Proc. Workshop on Information Security System Scoring and Ranking*. Applied Computer Security Associates, http://www.acsac.org/measurement/proceedings/wisssr1-proceedings.pdf

Alger, J. (2001) On Assurance, Measures, and Metrics: Definitions and Approaches. http://philby.ucsd.edu/~cse291_IDVA/papers/rating-position/Alger.pdf

Alves-Foss, J., & Barbosa, S. (1995). Assessing Computer Security Vulnerability. Operating Systems Review, Vol 29, No 3, July 1995, pp. 3–13.

Anderson, R. (2001). Security engineering: A guide to building dependable distributed systems, Wiley.

Andersson, R., Hunstad, A., & Hallberg, J. (2003). Evaluation of the security of components in distributed information systems. Linköping, Scientific report. FOI-R--1042—SE. Swedish Defence Research Agency.

Blobel, B. & Roger-France, F. (2001). A systematic approach for analysis and design of secure health information systems, International Journal of Medical Informatics, Vol. 62, Issue 1, pp. 51-78, June 2001.

Bond, A. & Påhlsson, N. (2004). A Quantitative Evaluation Framework for Component Security in Distributed Information Systems. Master's Thesis. LITH-ISY-EX-3574-2004. Linköpings universitet.

CC (1999). *Common Criteria for Information Technology Security Evaluation. Part 1: Introduction and general model, Part 2: Security functional requirements, Part 3: Security assurance requirements. Version 2.1*, August 1999.

Clark, K., Tyree, S., Dawkins, J., & Hale, J. (2004). Qualitative and Quantitative Analytical Techniques for Network Security Assessment. Proceedings of the 5th IEEE Workshop on Information Assurance. West Point, NY, June 2004.

ComputerWire. (2004). *Survey adds fuel to Linux/Windows security debate*. Computer Business Review Online. Available from: www.cbronline.com/article_news.asp?guid=FDFE64FB-2FEF-4952-A062-B60760AF9109.

Eschelbeck, G. (2000). Active Security – A proactive approach for computer security systems. *Journal of Network and Computer Applications*, Vol. 23, No.2, April 2000, pp. 109-130.

Gollmann, D. (1999). *Computer Security*. John Wiley & Sons.

Greenwald, M. & Gunter, C. (2003), *Computer security is not a science*, Large-Scale Network Security Workshop, Landsdowne, VA.

Hallberg J. (2000). *Secure User and System Authentication – Beyond Conventional Smart Cards*, Scientific Report, FOA-R--00-01495-505--SE, Defence Research Establishment, Linköping, Sweden.

Hunstad, A. & Hallberg, J. (2002). Design for securability - Applying engineering principles to the design of security architectures. ACSA workshop on the application of engineering

principles to system security design. Boston, Nov. 6-8, 2002. Linköping, FOI 2002, 8 p. (FOI-S--0721--SE)

Hunstad, A. & Hallberg, J. (2002b), *Modeling of Distributed Systems Focusing on IT Security Aspects*, FOI-R—0712-SE, FOI, Linköping, Sweden.

ITSEC. (1991). *Information Technology Security Evaluation Criteria*. Commission of the European Communities, Version 1.2.

Kovacich, G. (1997). Information Systems Security Metrics Management. *Computers & Security*, Vol. 16 (1997), No. 7, pp. 610-618

Leach, J. (2004). TBSE – an engineering approach to the design of accurate and reliable security systems. Computers & Security, Volume 23, Issue 1, February 2004, pp. 22-28.

Leung, H. (2001). Quality metrics for intranet applications. Information & Management 38 (2001). Pages 137-152.

NIST, National Institute of Standards and Technology. (1999). Controlled Access Protection Profile. October 1999. http://www.niap.nist.gov/cc-scheme/pp/PP_CAPP_V1.d.pdf

NSA, National Security Agency. (2001). *Protection Profile for Single-level Operating Systems in Environments Requiring Medium Robustness* (PP_MLOSPP-MR), Version 1.22

Oman, P., Krings, A., Conte de Leon, D., & Alves-Foss, J. (2004). Analyzing the Security and Survivability of Real-time Control Systems. Proceedings of the 5th IEEE Workshop on Information Assurance. West Point, NY, June 2004.

Oracle Corporation. (2004). Red Hat Enterprise Linux 3: Security Target, Version 1.7.

Payne, S. (2001). A Guide to Security Metrics. SANS Security Essentials GSEC Practical Assignment. http://www.sans.org/rr/whitepapers/auditing/55.php

Peterson, M. (2004). CAESAR - A proposed method for evaluating security in component-based distributed information systems. Master's Thesis. LITH-ISY-EX-3581-2004. Linköpings universitet.

Roberts, F., (1979). *Measurement Theory with Applications to Decision-making, Utility, and the Social Sciences*. Addison-Wesley.

Schudel, G. & Wood, B. (2000). Adversary Work Factor as a Metric for Information Assurance. *Proceedings of the New Security Paradigms Workshop*. Cork, Ireland, Sep. 18-22, 2000.

Schwartau, W. (2001). Network Security It's About Time: An Offer for a Metric. Network Security, Volume 2001, Issue 8, August 2001, pp. 11-13.

Science Applications International Corporation. (2002). Windows 2000 Security Target. ST Version 2.0. October 2002. http://www.niap.nist.gov/cc-scheme/st/ST_VID4002-ST.pdf

Secmet (2004). Security metrics consortium. WWW page (visited 2004-09-10). http://www.secmet.org/

Siponen, M. (2002) Designing Secure Information Systems and Software. Critical evaluation of the existing approaches and a new paradigm. Dept Information Processing Science and Infotech, Oulu, University of Oulu, Oulu 2002 A 387, ISBN 951-42-6789-3.

Swanson, M., Bartol, N., Sabato, J., & Hash, J. (2003). Security metrics guide for information technology systems. Technical Report NIST Special Publication 800-55, NIST, July 2003. http://csrc.nist.gov/publications/nistpubs/800-55/sp800-55.pdf.

TCSEC. (1983). Trusted Computer System Evaluation Criteria, (The Orange Book). US Department of Defense.

Vaughn, R., Henning, R., & Siraj, A. (2003). Information Assurance Measures and Metrics – State of Practice and Proposed Taxonomy. Proceedings of the Hawaii International Conference on System Sciences (HICSS-36), Waikoloa, Hawaii, January 6-9, 2003.

Wang, C. & Wulf, W. (1997). A Framework for Security Measurement. Proceedings of the National Information Systems Security Conference, Baltimore, MD, pp. 522-533, Oct. 1997.

Whitmore, J.J. (2001). A method for designing secure solutions. IBM Systems Journal, Armonk 2001, Volume 40, Issue 3.

Wood, B. & Bouchard, J. (2001). Red Team Work Factor as a Security Measurement. ACSA Workshop on Information Security System Rating and Ranking, Williamsburg, Virginia, 21-23 May 2001. http://philby.ucsd.edu/~cse291_IDVA/papers/rating-position/Bouchard.pdf

APPENDIX A
Caesar – a System Evaluation Method

This appendix presents a method for system security assessment called *CAESAR*[4] and is based on (Peterson, 2004), where CAESAR was introduced. The purpose of CAESAR is to illustrate the possibilities of structural system security assessment methods. Thus, it estimates the security level of an entire distributed information system, based on the security level of, and the relations between, its included components. However, being an illustration of structural system security assessment methods CAESAR is far from complete and the design choices made are topics of discussion.

CAESAR consists of two main parts:

- A modeling technique
- An evaluation algorithm

Figure 17 illustrates how the modeling technique and the evaluation algorithm together produce the overall security level of the distributed information system. Firstly, the distributed information system has to be modeled, using the modeling technique described below, data of the real system, and previously made security evaluations of components. The system model is then supplied to the evaluation algorithm described below, which calculates the overall security level of the system.

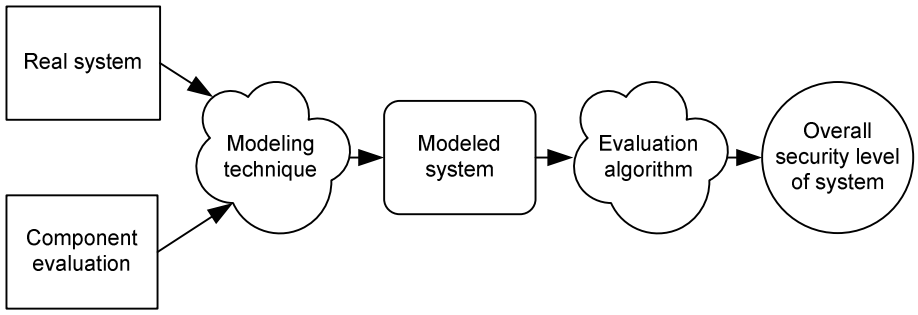


Figure 17: General workflow of CAESAR.

Modeling Technique

The main purpose of the modeling technique is to capture characteristics of the distributed information system that are important to its overall security level. The modeling technique consists of several building blocks. A brief overview of these

[4] Component-based Approach to Estimating the level of IT Security of Architecturally Rendered distributed information systems (CAESAR)

blocks and how they relate is presented in Figure 18. A modeled system consists of other modeled systems, system components, and component relations.

Component relations are physical relations, and logical relations. System components are traffic generators (computers and public networks) and traffic mediators (firewalls, routers, proxies and hubs). Properties in gray are virtual properties that are not allowed to be used in an actual model.

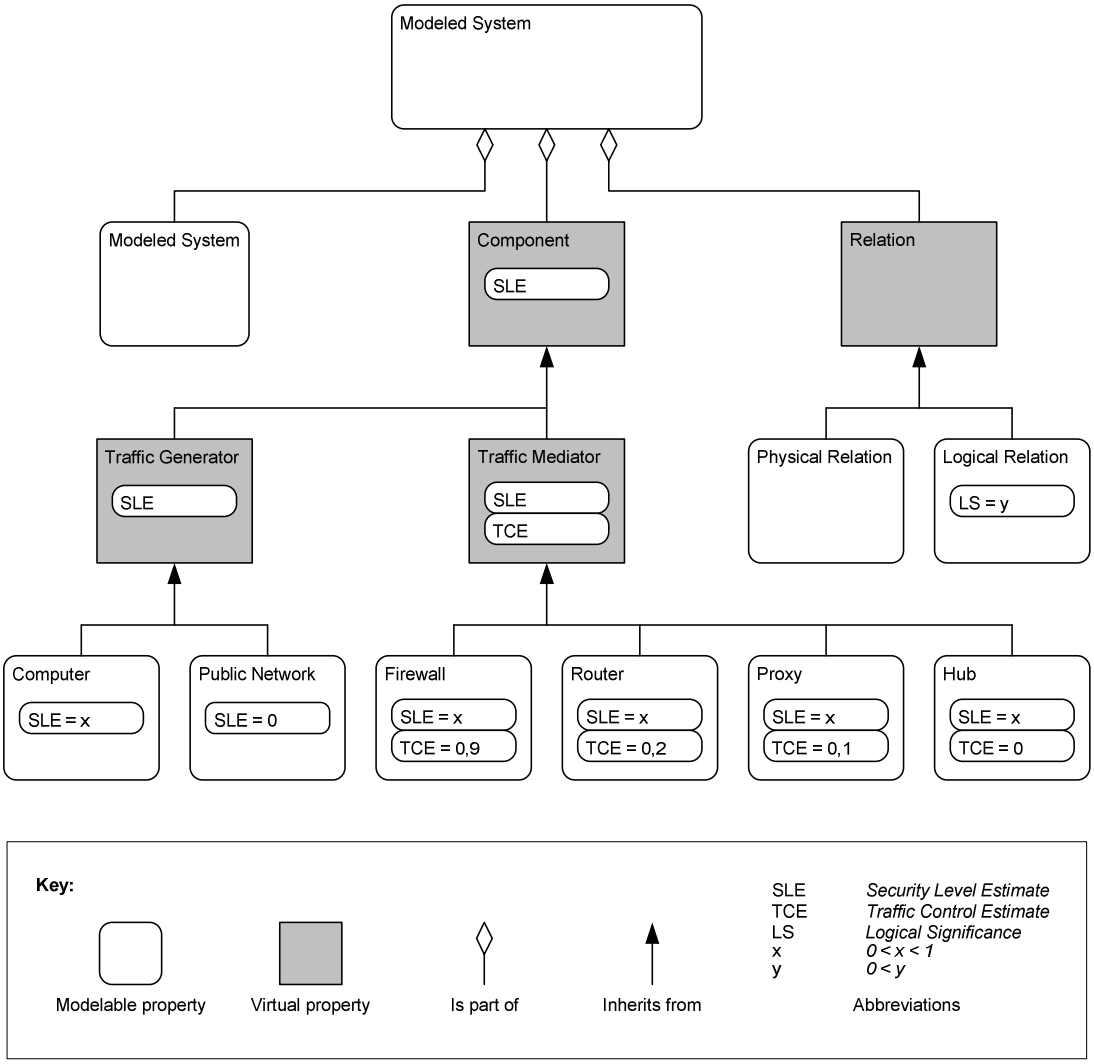


Figure 18: Building blocks of the CAESAR modeling technique.

Creating a model using the modeling technique of CAESAR consists of the following steps:

- identify all components,
- determine each component's class, that is, traffic generator or mediator,
- determine each component's security level estimate,
- determine each traffic mediator's traffic control estimate,
- determine physical relations between components, and

- determine logical relations between components.

The rest of this section will describe these concepts in detail.

Component classes

In order to create a model of a distributed information system, it is essential to define the smallest parts or components of which the modeled system is built – the atoms of the modeling technique.

It is important to decide on a finite number of such atoms, here called component classes. The number of component classes should be large enough for the resulting model to give a sufficiently detailed image of the system, but small enough to be unambiguous and comprehensible for the human user of the modeling technique.

It is crucial to understand that the number of component classes will determine the complexity of the evaluation algorithm that is to be applied to the modeled system later. An increased number of component classes will result in an increased complexity of the evaluation algorithm and vice versa.

The physical component classes are computer, public network, firewall, router, proxy, and hub. Some would argue that a class called Switch has been left out. In this thesis, a switch is considered equal to a hub from a security standpoint. If the user of CAESAR would be of a different opinion, it would be straightforward to add a component class.

These physical component classes are grouped into the super classes traffic generators (computers and public networks) and traffic mediators (firewalls, routers, proxies, and hubs). It is apparent that *traffic generators* separate from *traffic mediators* by their ability to generate traffic. In a simplified view, traffic generators could also generally be considered as security-decreasing components, while traffic mediators generally are security-increasing components. The functional difference between the traffic generators and traffic mediators as defined in this method will be explained in the following sections.

Security level estimate

Each component, corresponding to a component class, is also designated a *security level estimate* (SLE). The security level estimate may be calculated in a number of ways. However, no particular method for component evaluation has, this far, been adopted to CAESAR. This is an issue for future work. Here, the security level estimate is assumed to be on the form of a scalar value, which simplifies the description of CAESAR greatly. It would however, be equally possible to let the security level estimate, and therefore almost all other modeled and calculated

properties, to be on the form of a vector. The security level estimate is regarded as a ratio scale, regardless of what it is chosen to measure.

Traffic control estimate

Each traffic mediator is assigned a *traffic control estimate* (TCE) from 0 to 1, based on its ideal ability to conceal malicious network traffic. 0 indicates that the component has no flow control whatsoever, and 1 suggests that no malicious network traffic at all may pass the component.

One would of course like to think of the traffic control estimate as a ratio scale, and here it is defined as such, but to be truthful the measurement possibilities merely allows it to be an ordinal scale. The values are however treated as of ratio scale type when aggregated in the evaluation algorithm. It is imperative to recognize that the traffic control estimate is not an estimate of the actual function of a traffic mediator but rather of its purpose or ideal function.

Physical relations

Physical relations connect two components with each other. Physical relations contain no modeled properties, except references to the two connected components. Physical relations are considered symmetrical.

Logical relations

Logical relations connect two traffic generators with each other. The purpose introducing logical relations into the modeling technique is to be able to regard components' communicational premises and patterns and the implications of them. For example, with logical relations the evaluation algorithm takes into account that a server component, essential for many other components, may be more important than a client component, which no other components depend upon. Logical relations are non-symmetrical and contain a server-end and a client-end. Peer-to-peer relations are modeled as two logical relations; one in each direction.

Except the references to the two connected traffic generators and its direction, logical relations also contain a property, ranging from 0 to M, that denotes the significance of the relation. The upper bound M may be any value. If a logical significance of 5 is chosen for a component, its security level estimate will weigh five times that of a directly physically related component (six times, if the component is also both physically and logically related).

Evaluation Algorithm

The main goal of the algorithm is to calculate the *overall security level* (OSL) of the system, but many partial results are interesting in them selves and may be analyzed to understand how different factors affect the security level of different parts of the system.

To be able to calculate the overall security level of the system, it is necessary to perform an evaluation of each component of the traffic generator class. Such an evaluation results in a *system-dependent security level* (SSL) for that component.

Figure 19 gives a general overview of which modeled and calculated properties that are necessary to perform an evaluation of a system.

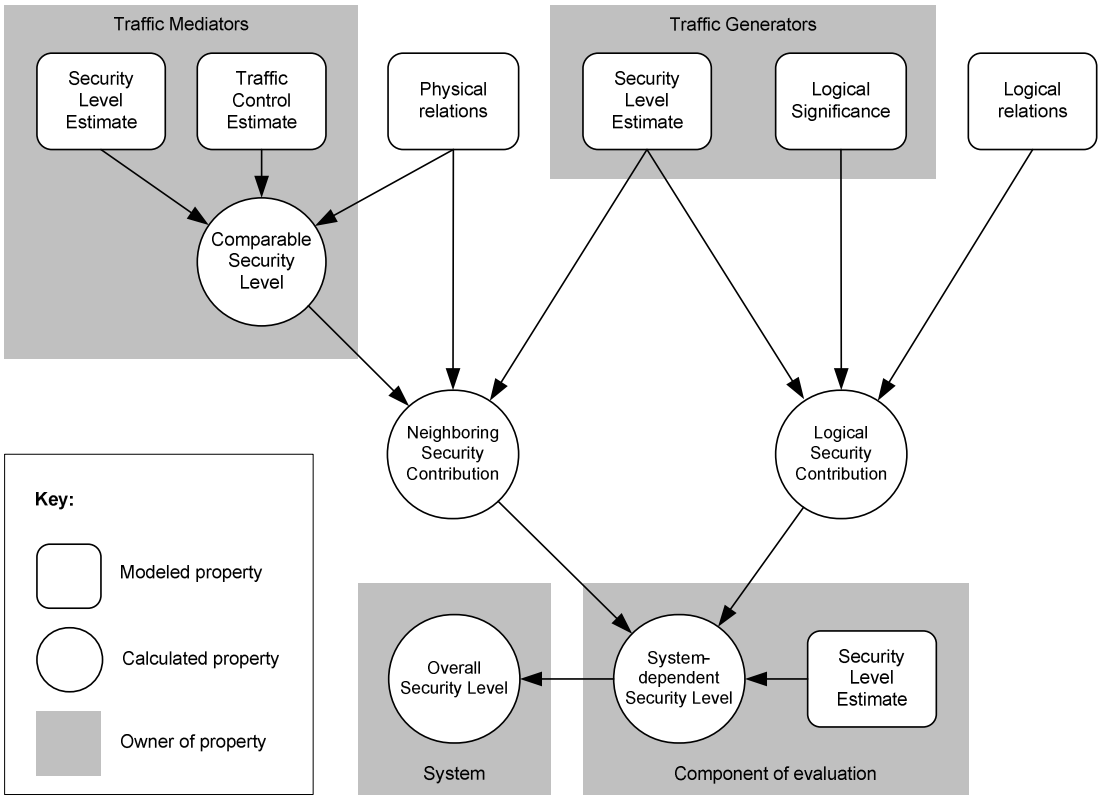


Figure 19: The CAESAR evaluation algorithm's main concepts and their relations.

The system-dependent security level of each traffic generator, from here on referred to as the component of evaluation, is calculated by aggregating the neighboring security contributions of all physically related traffic generators and the logical security contributions of all logically related traffic generators acting as servers to the component of evaluation.

The neighboring security contribution of a physically related traffic generator is calculated by aggregating the security level estimate of that traffic generator, with all traffic mediators that are on the physical path or paths between the traffic generator

and the component of evaluation taken into account by aggregating their comparable security level.

The logical security contribution of a logically related traffic generator acting as a server to the component of evaluation is calculated by aggregating the security level estimate of the traffic generator and the logical significance of the logical relation that connects the traffic generator with the component of evaluation.

The overall security level of the distributed system is then calculated by aggregating the system-dependent security level of all traffic generators.

The algorithm may also be described on a shorter form, as expressed below:

```
For each traffic generator, Ce, in system
        for each traffic generator, Cg, physically related to Ce
                calculate the aggregated CSL between Cg and Ce.
                calculate the NSC from Cg to Ce using CSL.
        for each traffic generator Cg, logically related to Ce
                calculate the LSC from Cg to Ce.
        calculate the SSL of Ce using all CSL, all LSC, and the SLE.
Calculate the security level of the system.
```

How these modeled and calculated properties are aggregated is described in the rest of this section.

Overall security level

The *overall security level* (OSL) of the system is the final result of CAESAR. It is calculated as an average of the system-dependent security level of all traffic generators in the system. Thus, the formula for a system with traffic generators, C_1, C_2, …, C_n is:

$$OSL(system) = \frac{SSL(C_1) + SSL(C_2) + ... + SSL(C_n))}{n}$$

System-dependent security level

In order to evaluate a component, further known as the component of evaluation, with respect to its environment – that is, calculating its system-dependent security level (SSL) – it is necessary to calculate the neighboring security contributions (NSC) of all other traffic-generating components that are physically related to the component of evaluation directly, or indirectly through a traffic-mediating component. It is also necessary to calculate the logical security contributions (LSC) of

all traffic-generating components that are logically related to the component of evaluation. Finally, it is necessary to know the security level estimate (SLE) of the component of evaluation. Figure 20, which is a subset of Figure 19, illustrates this graphically.

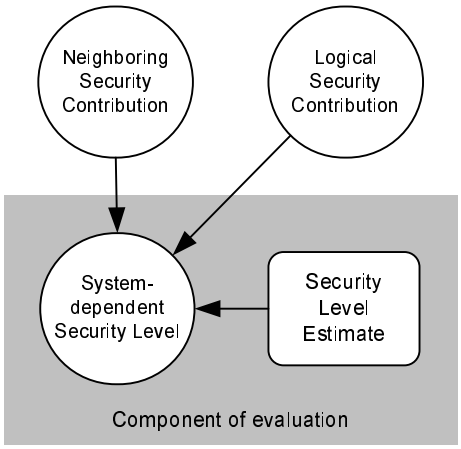


Figure 20: Factors that influence system-dependent security level.

When establishing which components that are allowed to give a neighboring security contribution to the component of evaluation, the following rule is used: All traffic generators that are directly or indirectly connected trough one or more physical relation(s) and possibly one or more traffic mediators should give a neighboring security contribution to the component of evaluation.

Thus, a traffic generator connected in series with a traffic generator connected in series with the component of evaluation does not give a neighboring security contribution to the component of evaluation. Traffic generators do not mediate traffic.

When determining which components that are allowed to give a logical security contribution to the component of evaluation, the rule is: All traffic generators that are logically related as servers should give a logical security contribution to the component of evaluation. Clients do not give any logical security contribution since the component of evaluation does not depend on them.

Figure 21 shows three examples of when neighboring security contribution (NSC) and logical security contribution (LSC) are used.

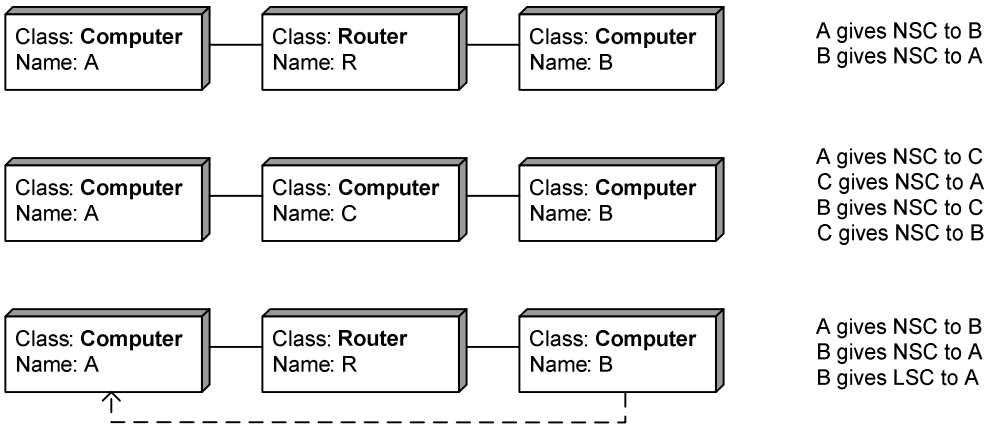


Figure 21: Examples showing when to use NSC and LSC.

The general formula for calculating the system-dependent security level of a component *Ce*, with respect to neighboring security contributions from C_{f1}, C_{f2}, …, C_{fn}, as well as logical security contributions from C_{l1}, C_{l2}, …, C_{lm}, whose logical significance are S_1, S_2, …, S_m respectively:

$$a = SLE(C_e) + NSC(C_{f1}) + NSC(C_{f2}) + ... + NSC(C_{fn}) +$$
$$+ LSC(C_{l1}) + LSC(C_{l2}) + ... + LSC(C_{lm})$$
$$b = 1 + n + (S_1 + S_2 + ... + S_m)$$
$$SSL(C_e) = \frac{a}{b}$$

The two following examples illustrate how to apply the general formula.

Example 1: Consider the simplest system; just two components, C_e and C_g, both traffic generators, connected with a physical relation. In order to calculate the system-dependent security level of C_e, we need to consider only the neighboring security contribution of C_g:

$$SSL(C_e) = \frac{SLE(C_e) + NSC(C_g)}{1+1}$$

Example 2: Now instead, consider the scenario where the component of evaluation, C_e, is connected with one other traffic generator, C_1, through a physical relation, and to two other traffic generators C_2 and C_3, whose logical significances are 3 and 4, through logical relations. In that case the system-dependent security level of C_e would be:

$$SSL(C_e) = \frac{SLE(C_e) + NSC(C_1) + LSC(C_2) + LSC(C_3)}{1+1+3+4}$$

Neighboring security contribution

The term *neighboring security contribution* (NSC) is introduced to take into account the security level of physically related components, and especially to regard the possible

threat a component with a low security level estimate poses to the component of evaluation.

Figure 22, which is a subset of Figure 19, explains which system properties that are used when calculating the neighboring security contribution of a component.

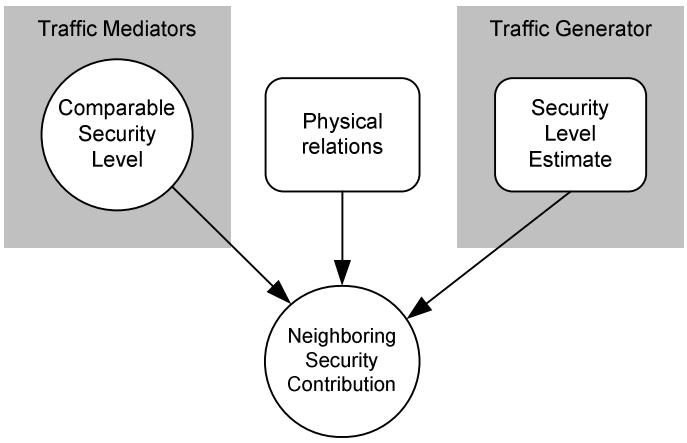


Figure 22: Factors that influence neighboring security contribution.

All traffic generators that are directly or indirectly connected trough one or more physical relations and possibly one or more traffic mediators should give a neighboring security contribution to the component of evaluation.

Consider the simplest system; just two components, C_e and C_g, both traffic generators, connected with a physical relation. In order to calculate the neighboring security contribution (NSC) of traffic generator C_g to component of evaluation *Ce*, the following expression is used:

$$NSC(C_g) = SLE(C_g)$$

That is, the neighboring security contribution of C_g equals its security level estimate (SLE).

If there is a traffic mediator, C_m, on the path between C_g and C_e, this must be taken into account, since the traffic mediator might filter out malicious traffic and therefore increase the neighboring security contribution of C_g.

If there is precisely one path from C_g to C_e and that path contains precisely one traffic mediator, C_m, the following expression gives the neighboring security contribution of C_g to C_e:

$$NSC(C_g) = \mathrm{Max}(CSL(C_m), SLE(C_g))$$

CSL stands for comparable security level, and is a measurement of how well a traffic mediator filters malicious traffic.

If there is more than one traffic mediator or several paths with traffic mediators between the traffic generator and the component of evaluation it is necessary to aggregate the comparable security level of the traffic generators into a single comparable security level, as described below. This value may then be used as above. The chosen expression for calculating neighboring security contributions is blunt, but delivers plausible estimations. There certainly are more elaborated expressions to be used.

Comparable security level

In order to calculate the neighboring security contribution of C_n to C_e, it is necessary to consider the combined effect of all traffic mediators on the paths between C_n and C_e. The concept of *comparable security level* (CSL) is introduced to describe the effect of traffic mediators. In Figure 23, which is a subset of Figure 19, the system properties used when calculating the comparable security level of a traffic mediator are illustrated.

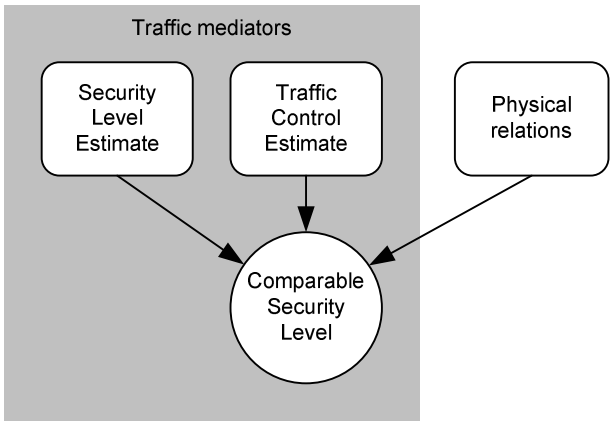


Figure 23: Factors that influence comparable security level.

The comparable security level of a single traffic mediator, C_m, is calculated by multiplying the component's traffic control estimate with its security level estimate:

$$CSL(C_m) = TCE(C_m) \cdot SLE(C_m)$$

Aggregating serial traffic mediators

If there is one path between C_n and C_e, but two traffic mediators, C_1 and C_2, are in series along that path, the following expression gives the aggregated comparable security level $CSL(C_a)$ for the traffic mediators, C_1 and C_2:

$$CSL(C_a) = \text{Max}(CSL(C_1), CSL(C_2))$$

This algebraic expression for aggregating traffic mediators in series assume that traffic mediators with a lower comparable security level are functional subsets of traffic mediators with a higher comparable security level. Considering the classes Firewall, Router, Proxy and Hub, that might be the case if these are provided with a security level estimate near 1. If they have a security level estimate considerable lower than 1, it might, however, not be an appropriate approximation.

If the traffic mediators C_1 and C_2 mentioned above instead would be considered entirely complementary, the following expression would describe the comparable security level of those traffic mediators:

$$CSL(C_a) = CSL(C_1) + (1 - CSL(C_1)) \cdot CSL(C_2)$$

These two cases are extreme ones that will virtually never appear in a realistically modeled distributed system. They might however still be useful as simplifications.

If the degree of overlapping functionality between C_1 and C_2 would be known as f, ranging from 0 to 1, it would be possible to describe the comparable security level with:

$$CSL(C_a) = f \cdot \mathrm{Min}(CSL(C_1), CSL(C_2)) + (1 - f) \cdot (CSL(C_1) + (1 - CSL(C_1)) \cdot CSL(C_2))$$

If there are more than two traffic mediators in series along a path, the equations may be extended to accept more than two traffic mediators as input, or the equations may be applied iteratively on pairs of traffic mediators.

Aggregating parallel traffic mediators

If there are precisely two paths between C_n and C_e and each path contains one traffic mediator, C_1 and C_2 respectively, the following expression returns the aggregated comparable security level $CSL(C_a)$ for the traffic mediators:

$$CSL(C_a) = \mathrm{Min}(CSL(C_1), CSL(C_2))$$

This approach assumes that a network is not stronger than its weakest link, and as long as the weaker component does not increase its security level estimate or is replaced with a component with a higher traffic control estimate, it does not matter how much the comparable security level of the stronger component is increased. However, since it is a network and not a chain, it is probably weaker than the weakest link.

If there are more than two paths, the equations may be applied iteratively on pairs of paths. If some paths contain more than one traffic mediator, these must first be aggregated, as described above, so that the path contains one aggregated traffic mediator only.

Logical security contribution

The concept of *logical security contribution* (LSC) is introduced to increase the sensitivity of the overall security level to the security level of traffic generators that are acting as servers to other components in the system. Thus, logical security contribution is constructed to reflect such differences in importance between different components.

The logical security contribution of a component is determined by its security level estimate, its logical relation to the component of evaluation, and the logical significance of that relation. Figure 24, which is a subset of Figure 19, illustrates this graphically.

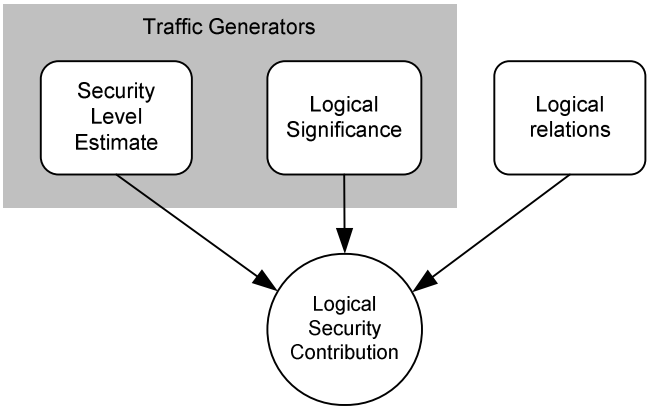


Figure 24: Factors that influence logical security contribution.

As previously mentioned, all traffic generators that are logically related as servers should give a logical security contribution to the component of evaluation. The logical security contribution (LSC) of a traffic generator, C_g, with a logical relation (LR) of a certain logical significance (LS), to the component of evaluation, C_e, is expressed by:

$$LSC(C_g) = LS(LR(C_g, C_e) \cdot SLE(C_g))$$

Characteristics of the Approach

CAESAR can be applied to both designed, but not implemented, and fully implemented systems. Thus, it can be used when laying out the overall structure of distributed information systems, but also an asset in the everyday monitoring of system modifications and improvements. The ability of CAESAR to regard traffic flow control seems to be a unique quality. No previous example of such a method could be found.

The simplicity of CAESAR is a strong point. It is both easy to use, and to implement in computer software. Moreover CAESAR has a modular structure. Thus, it is possible to alter a part of the modeling technique or evaluation algorithm without

affecting other parts. This makes CAESAR valuable as an instrument for evaluation of different expressions and principles for structural system security assessment.

In this portrayal of CAESAR, the security level estimate has been referred to as a scalar. This may not necessarily be the only way to represent security levels in CAESAR. However, the security level estimate could be multi-dimensional. In fact, most other modeled and calculated properties of CAESAR may very well be vectors.

APPENDIX B
The ROME Software

The software implementation of the CAESAR system security assessment method is called ROME. The purpose of ROME is threefold:

- to illustrate the CAESAR method with an intuitive and comprehensive tool,
- to evaluate existing or planned systems using the CAESAR method, and
- to evaluate and enhance the CAESAR method.

ROME supports all classes and physical relations as described in Appendix A. It allows system models to be created and stored as well as existing system models to be modified or examined. Models are evaluated in real time. The alterations of relations and components immediately affect other components and the overall security level. Evaluation results are displayed in highly configurable color or shape shifts, on either one of several component levels or on the overall system level.

Figure 25 shows a screenshot of the main window of the ROME software.

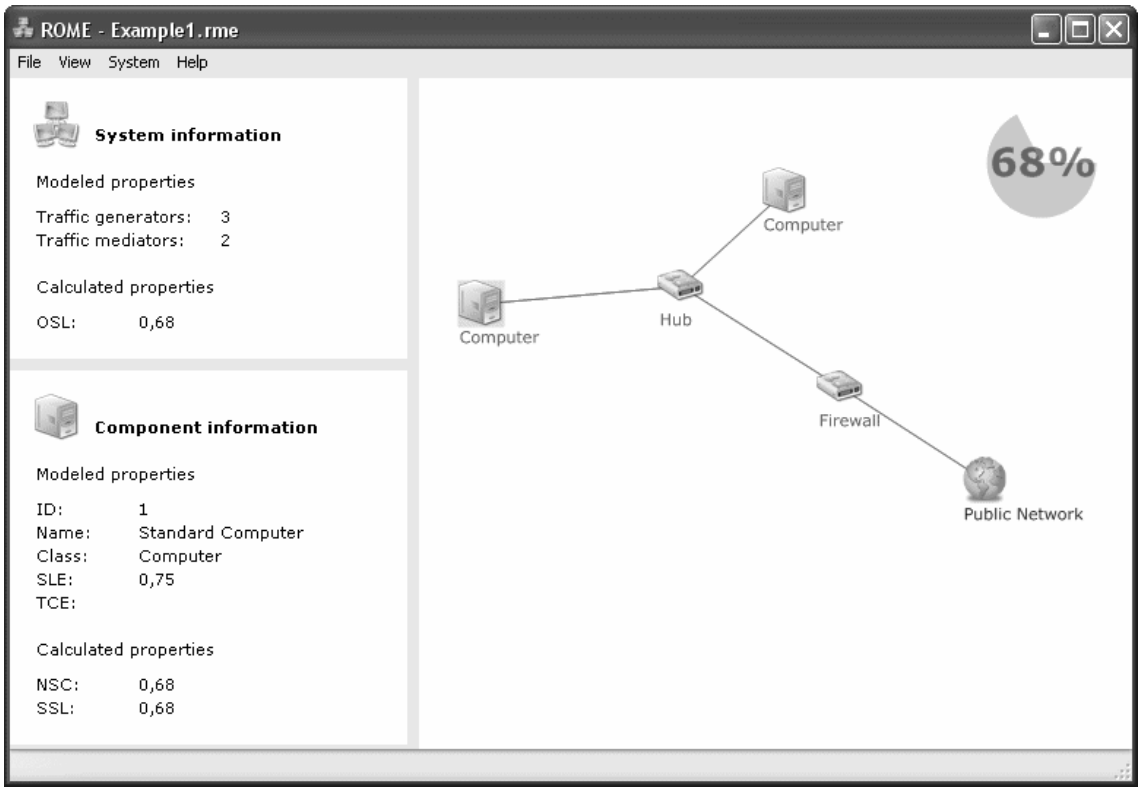


Figure 25: Screenshot of the ROME software.

In the upper left corner, system information is displayed. Both modeled properties, such as the number of components, and calculated properties, such as the overall

security level are displayed. In the lower left corner, the currently selected component is showed along with both modeled properties, such as its name, class and security level estimate, and calculated properties, such as its neighboring security contribution and system-dependent security level.

In the upper right corner, a pie chart along with a percentile is shown. The color of the pie chart changes depending on the current overall security level of the modeled system from blue when 100% secure to red when 0% secure.

The large window to the right is the workspace, where the modeled system is displayed. Each class has its own icon. Physical relations are shown as gray lines between components.

When first starting the program, it is possible to load an existing system, or create a new system. To create a new system, click with the right mouse button in the large space to the right, select Add > Standard Component > Computer, as shown in Figure 26. It is possible to add supplementary components in the same manner. Physical relations are created by clicking on a component with the right mouse button, selecting Add > Relation > Physical relation, and then clicking with the left mouse button on another component.

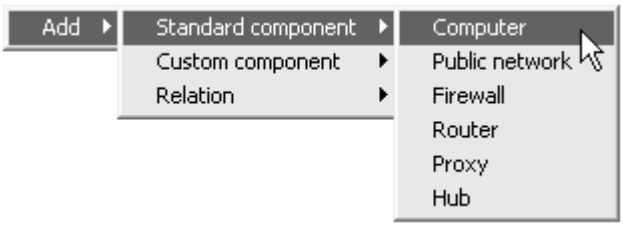


Figure 26: Adding components using the ROME software.

The texts and colors used to represent the system in the workspace are configurable. For example, to let the color of the text label of each component represent its system-dependent security level ranging from blue when 100% secure to red when 0% secure, click on the menu View > Node color > System-dependent Security Level (SSL) as shown in Figure 27.

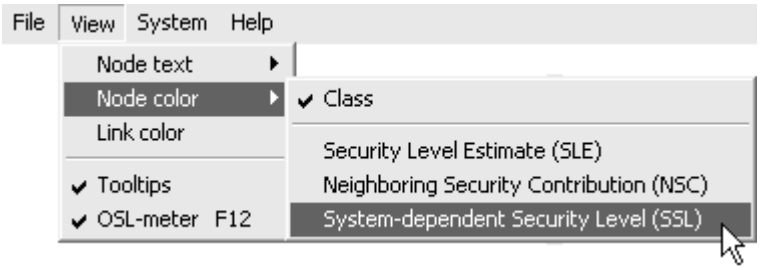


Figure 27: Adjusting representation using the ROME software.

Figure 28 shows what the workspace might look like using these settings to represent the system-dependent security level of each component.

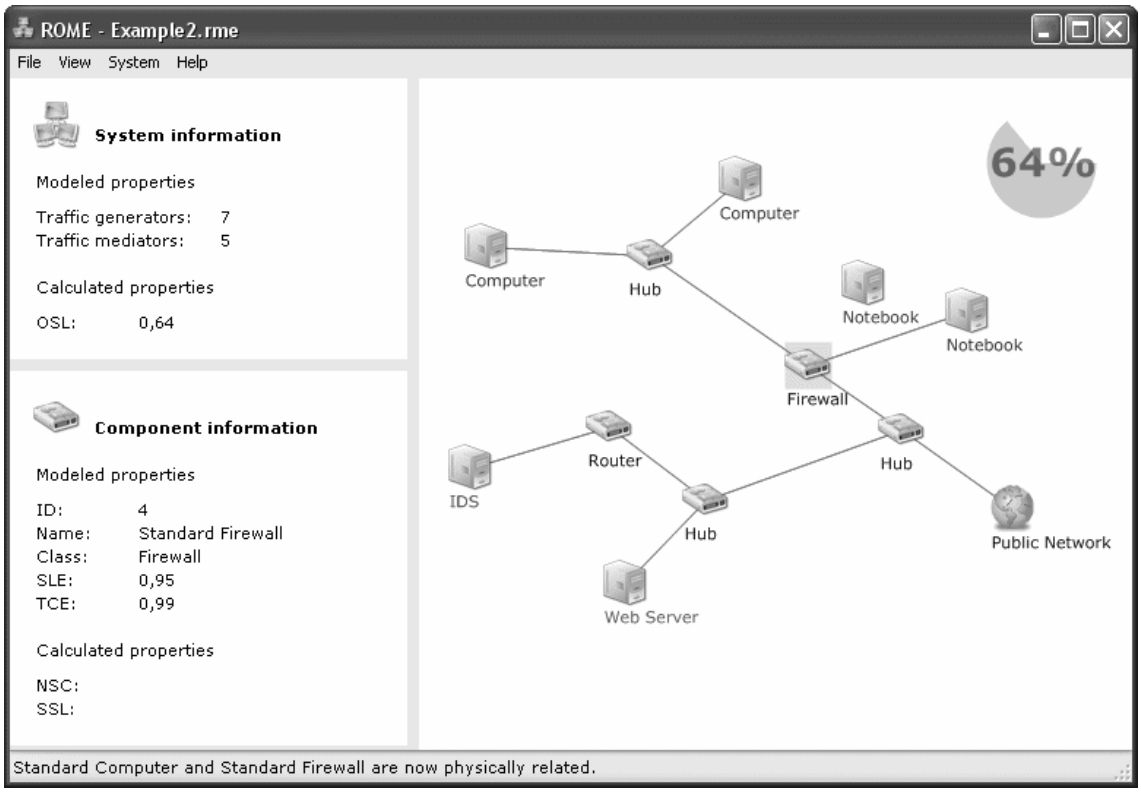


Figure 28: An example of a modeled system using the ROME software.

APPENDIX C
Common Criteria

Common Criteria (CC, 1999) is a widely spread and accepted evaluation method, originating from its predecessors TCSEC (1983) and ITSEC (1991). It is based on a set of standardized Security Functional Requirements (SFR) that can be expressed in Protection Profiles (PP) and Security Targets (ST). The product, which the latter describes the behavior of, is referred to as the Target of Evaluation (TOE). CC is divided into three parts. Part 1 includes a general introduction and overview of CC. Part 2 provides a language for functional descriptions of security requirements and TOE security functions. Part 3 describes the assurance requirements of CC, that is, what is required to achieve a certain Evaluation Assurance Level (EAL).

The SFRs are divided into eleven classes, each describing different security aspects. These classes are further divided into families, which in turn consist of components. The components can be made up of one or more elements. In Figure 29, the first family contains three hierarchical components, where component 2 and component 3 can both be used to satisfy dependencies on component 1. Component 3 is hierarchical to component 2 and can also be used to satisfy dependencies on component 2. (CC, 1999)

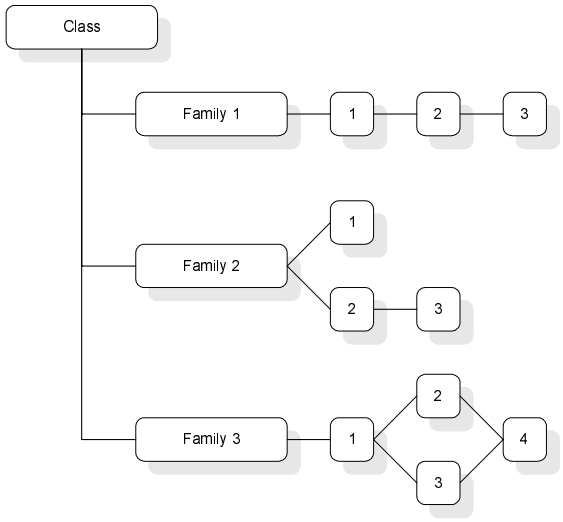


Figure 29: Class decomposition diagram (CC, 1999).

In the second family there are three components, of which not all are hierarchical. Components 1 and 2 are hierarchical to no other components. Component 3 is hierarchical to component 2, and can be used to satisfy dependencies on component 2, but not to satisfy dependencies on component 1.

In the third family, components 2, 3, and 4 are hierarchical to component 1. Components 2 and 3 are both hierarchical to component 1, but non-comparable. Component 4 is hierarchical to both component 2 and component 3.

The SFRs are divided into the following eleven classes (CC, 1999):

- **FAU – Security Audit**
  Security auditing involves recognising, recording, storing, and analysing information related to security relevant activities
- **FCO – Communication**
  The FCO class provides two families specifically concerned with assuring the identity of a party participating in a data exchange. These families ensure that an originator cannot deny having sent the message, nor can the recipient deny having received it.
- **FCS – Cryptographic Support**
  The TOE Security Functions may employ cryptographic functionality to help satisfy several high-level security objectives. This class is used when the TOE implements cryptographic functions, the implementation of which could be in hardware, firmware and/or software.
- **FDP – User Data Protection**
  The FDP class contains families specifying requirements for TOE security functions and TOE security function policies related to protecting user data. FDP is split into four groups of families that address user data within a TOE, during import, export, and storage as well as security attributes directly related to user data.
- **FIA – Identification and Authentication**
  The families in the FIA class deal with determining and verifying the claimed identity of users, determining their authority to interact with the TOE, and with the correct association of security attributes for each authorised user.
- **FMT – Security Management**
  The FMT class is intended to specify the management of several aspects of the TSF: security attributes, TSF data, and functions. The different management roles and their interaction, such as separation of capability, can be specified.
- **FPR – Privacy**
  The FPR class contains privacy requirements. These requirements provide a user protection against discovery and misuse of identity by other users.
- **FPT – Protection of the TSF**
  The FPT class contains families of functional requirements that relate to the integrity and management of the mechanisms that provide the TOE security functions and to the integrity of its data.
- **FRU – Resource Utilisation**
  The FRU class support the availability of required resources such as processing capability and/or storage capacity.
- **FTA – TOE Access**
  The FTA class specifies functional requirements for controlling the establishment of a user's session.

- **FTP – Trusted Path/Channels**
  Families in this class provide requirements for a trusted communication path between users and the TOE security functions, and for a trusted communication channel between the TOE security functions and other trusted IT products.

A PP specifies a profile of the implementation-independent requirements for a category of products or systems that meet specific customer needs, whereas a ST specifies the implementation-dependent security functionality used as a basis for a particular product or system. In Figure 30 and Figure 31, the structure of the PP and ST documents are presented respectively.

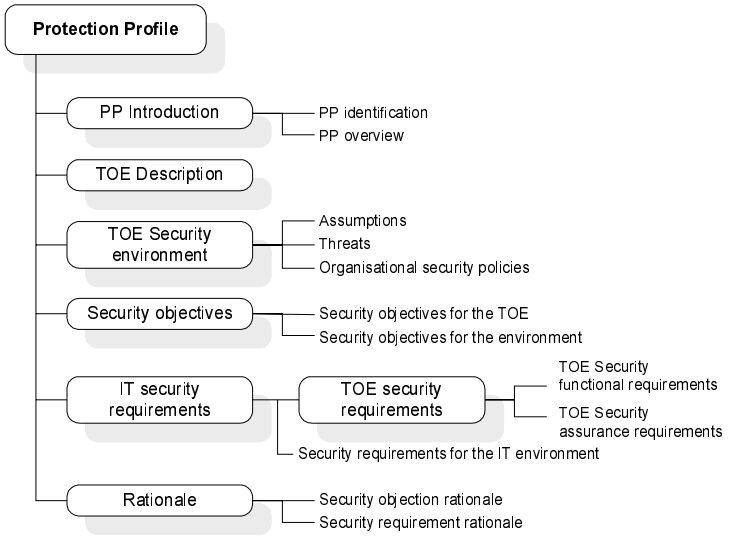


Figure 30: Specifications of Protection Profile (PP) (CC, 1999).

PPs can be developed using the methods of CC. To simplify this process, CC has a catalogue of standard SFRs which holds a set of functional components used to express functional requirements of products and systems.

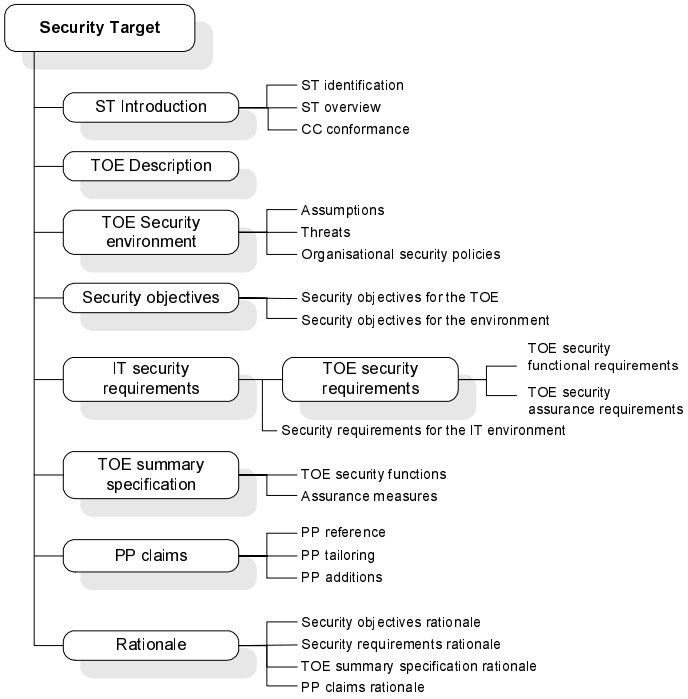


Figure 31: Specifications of a Security Target (ST) (CC, 1999).

A CC evaluation is carried out against a set of predefined assurance levels, Evaluation Assurance Levels (EAL1 to EAL7). These levels represent the ascending level of trust that can be placed in the implementation of the security functionality of the TOE.

APPENDIX D
Heimdal – Applying a Component Evaluation Framework

An overview of the Heimdal Framework is given here using a few different products as test cases. For further details regarding the framework, the interested reader is referred to (Bond and Påhlsson, 2004). The actual evaluation is made using Heimdal Security Evaluator 3000 .NET, a Windows application developed within the project. The program is explained in greater detail in Appendix E and in (Bond and Påhlsson, 2004).

Heimdal Security Evaluator 3000 .NET

The software used for the evaluations in this chapter is Heimdal Security Evaluator 3000 .NET (Figure 32). It is based on the Heimdal Framework, its profiles and algorithms presented in chapter 6 and in detail in .

A user may create and save all profiles required to perform an evaluation. These may then be combined to generate the actual evaluation.

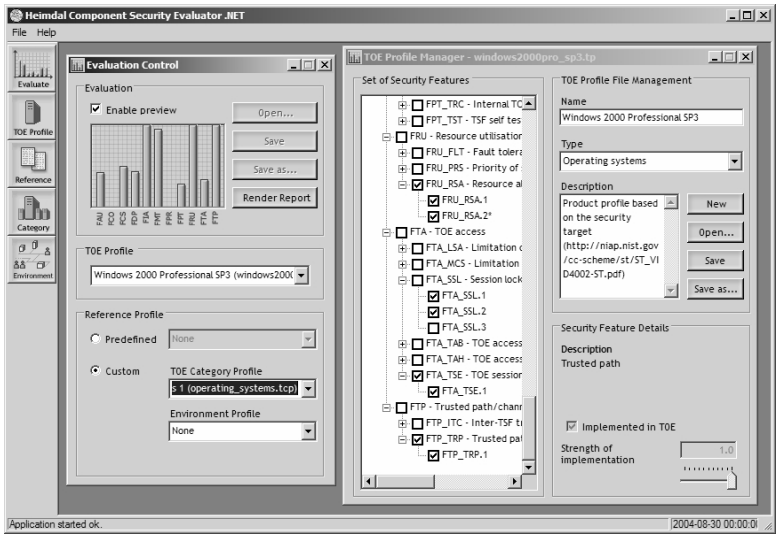
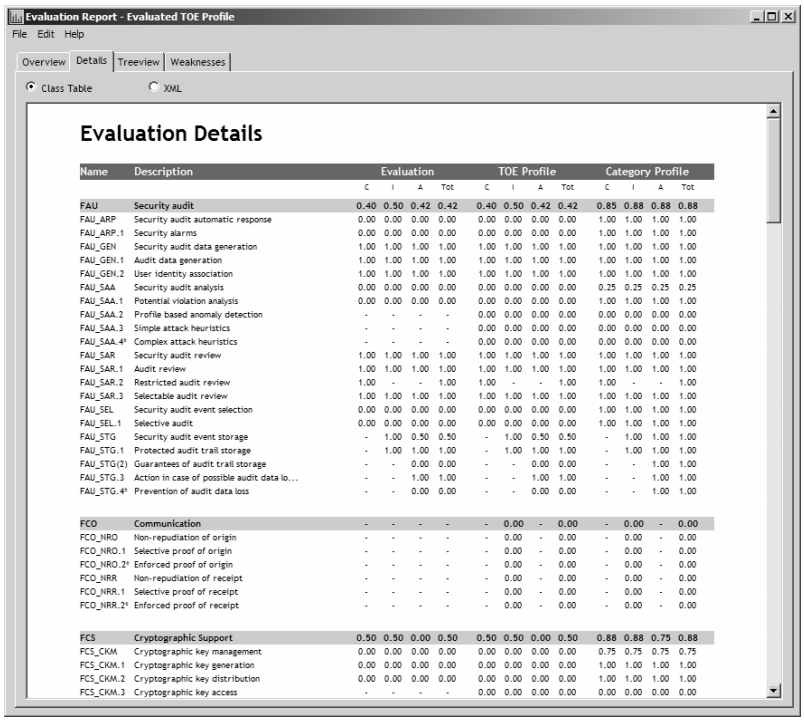


Figure 32: Security Evaluator main window.

The evaluator can choose to view the evaluation report either as bars on the Security class level, or in more details – as a table containing the security values at all levels. The latter is displayed below in Figure 33.

The evaluation report may be exported to Microsoft Excel or XML formats.



Name	Description	Evaluation				TOE Profile				Category Profile			
		C	I	A	Tot	C	I	A	Tot	C	I	A	Tot
FAU	Security audit	0.40	0.50	0.42	0.42	0.40	0.50	0.42	0.42	0.85	0.88	0.88	0.88
FAU_ARP	Security audit automatic response	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	1.00	1.00
FAU_ARP.1	Security alarms	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	1.00	1.00
FAU_GEN	Security audit data generation	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
FAU_GEN.1	Audit data generation	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
FAU_GEN.2	User identity association	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
FAU_SAA	Security audit analysis	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.25	0.25	0.25	0.25
FAU_SAA.1	Potential violation analysis	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	1.00	1.00
FAU_SAA.2	Profile based anomaly detection	-	-	-	-	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FAU_SAA.3	Simple attack heuristics	-	-	-	-	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FAU_SAA.4[a]	Complex attack heuristics	-	-	-	-	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FAU_SAR	Security audit review	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
FAU_SAR.1	Audit review	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
FAU_SAR.2	Restricted audit review	1.00	-	-	1.00	1.00	-	-	1.00	1.00	-	-	1.00
FAU_SAR.3	Selectable audit review	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
FAU_SEL	Security audit event selection	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	1.00	1.00
FAU_SEL.1	Selective audit	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	1.00	1.00
FAU_STG	Security audit event storage	-	1.00	0.50	0.50	-	1.00	0.50	0.50	-	1.00	1.00	1.00
FAU_STG.1	Protected audit trail storage	-	1.00	1.00	1.00	-	1.00	1.00	1.00	-	1.00	1.00	1.00
FAU_STG(2)	Guarantees of audit trail storage	-	-	0.00	0.00	-	-	0.00	0.00	-	-	1.00	1.00
FAU_STG.3	Action in case of possible audit data lo...	-	-	1.00	1.00	-	-	1.00	1.00	-	-	1.00	1.00
FAU_STG.4[a]	Prevention of audit data loss	-	-	0.00	0.00	-	-	0.00	0.00	-	-	1.00	1.00
FCO	Communication	-	-	-	-	-	0.00	-	0.00	-	0.00	-	0.00
FCO_NRO	Non-repudiation of origin	-	-	-	-	-	0.00	-	0.00	-	0.00	-	0.00
FCO_NRO.1	Selective proof of origin	-	-	-	-	-	0.00	-	0.00	-	0.00	-	0.00
FCO_NRO.2[a]	Enforced proof of origin	-	-	-	-	-	0.00	-	0.00	-	0.00	-	0.00
FCO_NRR	Non-repudiation of receipt	-	-	-	-	-	0.00	-	0.00	-	0.00	-	0.00
FCO_NRR.1	Selective proof of receipt	-	-	-	-	-	0.00	-	0.00	-	0.00	-	0.00
FCO_NRR.2[a]	Enforced proof of receipt	-	-	-	-	-	0.00	-	0.00	-	0.00	-	0.00
FCS	Cryptographic Support	0.50	0.50	0.00	0.50	0.50	0.50	0.00	0.50	0.88	0.88	0.75	0.88
FCS_CKM	Cryptographic key management	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.75	0.75	0.75	0.75
FCS_CKM.1	Cryptographic key generation	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	1.00	1.00
FCS_CKM.2	Cryptographic key distribution	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	1.00	1.00
FCS_CKM.3	Cryptographic key access	-	-	-	-	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Figure 33: Security Evaluator 3000 Report window (Details).

Example 1 – Windows 2000 Professional

Microsoft Windows 2000 Professional with Service Pack 3 and Q326886 Hotfix installed is used as the first test case. The development of a TOE Profile will be explained. Next, the TOE Category Profile will be developed and applied. Finally, the Reference Profile and Environment Profile should be developed and applied, resulting in a complete evaluation, but for details regarding this, the interested reader is referred to (Bond and Påhlsson, 2004).

Developing a TOE Profile – Windows 2000 Professional

The TOE Profile for Windows 2000 Professional was developed using the ST produced by Science Applications International Corporation (2002) as an input to the evaluation software. The Security Features stated to be implemented are given the value 1, the others 0.

The resulting TOE Profile will show the characteristics displayed in Figure 34. It shows the total security values on class level for each Security Class, and the confidentiality, integrity and availability values respectively.

According to the ST, the TOE Profile has no implemented Security Features in the FCO and FPR Security Classes, which are hence given the Security Value 0.

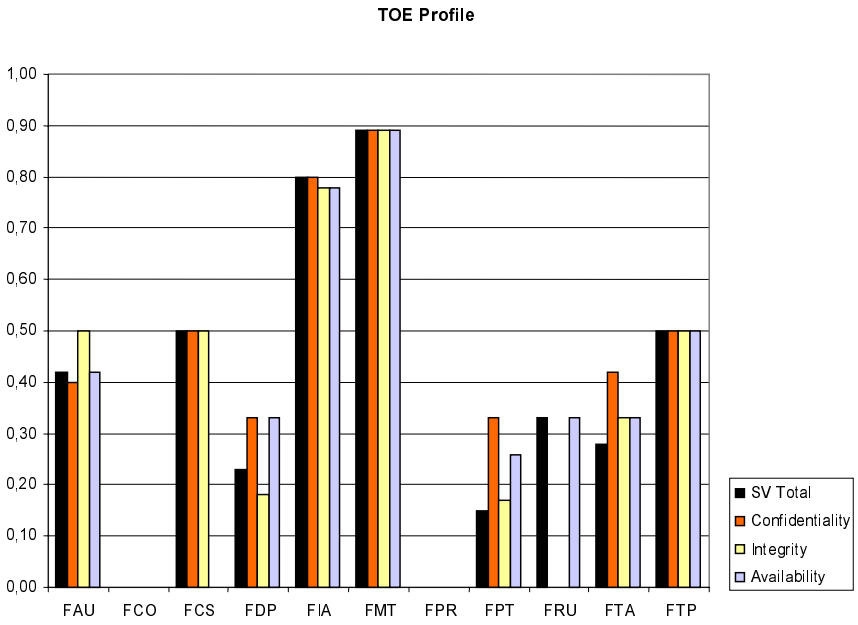


Figure 34: The actual TOE Profile for Windows 2000 Professional.

More details on the development process for a TOE Profile are found in Section 6.3.

Evaluating the TOE Profile with a TOE Category Profile

The next step in the evaluation is to create a TOE Category Profile (TCP) for operating systems.

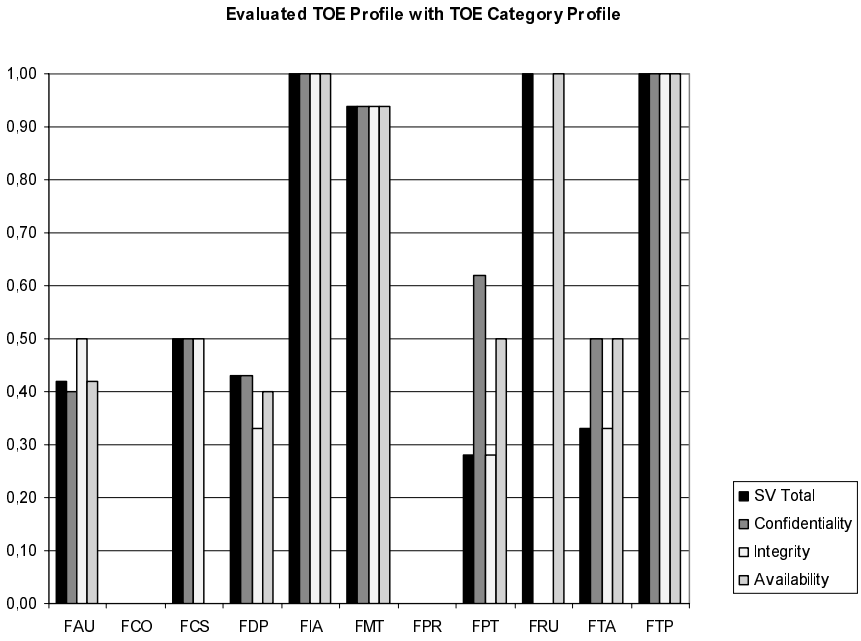


Figure 35: The Windows 2000 Professional TOE Profile evaluated with the TOE Category Profile for operating systems.

Applying the TCP will scale the values according to what functional requirements are considered important and relevant, i.e. those included in the TCP. The resulting values are expressed in an Evaluated TOE Profile. Figure 35 shows the characteristics of the Evaluated TOE Profile.

Comparing the TOE Profile in Figure 34 to the Evaluated TOE Profile in Figure 35, some class Security Values have been increased in the latter, while some are left unaffected. The ones increased have been so because the requirements from the TOE Category Profile are lower for those Security Classes.

Example 2 – Comparing Linux and Windows 2000

The modular design of the framework proposed in the previous chapter enables the comparison of products within the same product categories. This is exemplified in this section by comparing the Windows 2000 TOE Profile used in Section 5.2 to Red Hat Enterprise Linux, using the same TOE Category Profile as in Section 5.2. The Red Hat TOE Profile was developed in the same way as the Windows 2000 TOE Profile, using the Red Hat ST (Oracle, 2004). The results from the evaluation are presented in Table 5.

Table 5: Comparison between Windows 2000 and Red Hat Enterprise Linux evaluations.

Name	Windows 2000					Linux 3			
	C	I	A	Tot		C	I	A	Tot
FAU	0.40	0.50	0.42	0.42		0.40	0.50	0.50	0.50
FAU_ARP	0.00	0.00	0.00	0.00		0.00	0.00	0.00	0.00
FAU_GEN	1.00	1.00	1.00	1.00		1.00	1.00	1.00	1.00
FAU_SAA	0.00	0.00	0.00	0.00		0.00	0.00	0.00	0.00
FAU_SAR	1.00	1.00	1.00	1.00		1.00	1.00	1.00	1.00
FAU_SEL	0.00	0.00	0.00	0.00		0.00	0.00	0.00	0.00
FAU_STG	-	1.00	0.50	0.50		-	1.00	1.00	1.00
FCO	-	-	-	-		-	-	-	-
FCO_NRO	-	-	-	-		-	-	-	-
FCO_NRR	-	-	-	-		-	-	-	-
FCS	0.50	0.50	0.00	0.50		0.50	0.50	0.00	0.50
FCS_CKM	0.00	0.00	0.00	0.00		0.00	0.00	0.00	0.00
FCS_COP	1.00	1.00	-	1.00		1.00	1.00	-	1.00
FDP	0.43	0.33	0.40	0.43		0.43	0.33	0.40	0.43
FDP_ACC	1.00	1.00	1.00	1.00		1.00	1.00	1.00	1.00
FDP_ACF	1.00	1.00	1.00	1.00		1.00	1.00	1.00	1.00
FDP_DAU	-	-	-	-		-	-	-	-
FDP_ETC	-	-	-	-		-	-	-	-
FDP_IFC	0.00	0.00	0.00	0.00		0.00	0.00	0.00	0.00
FDP_IFF	0.00	0.00	0.00	0.00		0.00	0.00	0.00	0.00
FDP_ITC	0.00	0.00	0.00	0.00		0.00	0.00	0.00	0.00
FDP_ITT	0.00	0.00	-	0.00		0.00	0.00	-	0.00
FDP_RIP	1.00	-	-	1.00		1.00	-	-	1.00
FDP_ROL	-	-	-	-		-	-	-	-
FDP_SDI	-	-	-	-		-	-	-	-
FDP_UCT	-	-	-	-		-	-	-	-

Name	Windows 2000					Linux 3			
	C	I	A	Tot		C	I	A	Tot
FDP_UIT	-	-	-	-		-	-	-	-
FIA	1.00	1.00	1.00	1.00		0.83	0.83	0.83	0.83
FIA_AFL	1.00	1.00	1.00	1.00		0.00	0.00	0.00	0.00
FIA_ATD	1.00	1.00	1.00	1.00		1.00	1.00	1.00	1.00
FIA_SOS	1.00	1.00	1.00	1.00		1.00	1.00	1.00	1.00
FIA_UAU	1.00	1.00	1.00	1.00		1.00	1.00	1.00	1.00
FIA_UID	1.00	1.00	1.00	1.00		1.00	1.00	1.00	1.00
FIA_USB	1.00	1.00	1.00	1.00		1.00	1.00	1.00	1.00
FMT	0.94	0.94	0.94	0.94		0.56	0.56	0.56	0.56
FMT_MOF	1.00	1.00	1.00	1.00		0.00	0.00	0.00	0.00
FMT_MSA	0.67	0.67	0.67	0.67		0.67	0.67	0.67	0.67
FMT_MTD	1.00	1.00	1.00	1.00		1.00	1.00	1.00	1.00
FMT_REV	1.00	1.00	1.00	1.00		1.00	1.00	1.00	1.00
FMT_SAE	1.00	1.00	1.00	1.00		0.00	0.00	0.00	0.00
FMT_SMR	1.00	1.00	1.00	1.00		0.67	0.67	0.67	0.67
FPR	-	-	-	-		-	-	-	-
FPR_ANO	-	-	-	-		-	-	-	-
FPR_PSE	-	-	-	-		-	-	-	-
FPR_UNL	-	-	-	-		-	-	-	-
FPR_UNO	-	-	-	-		-	-	-	-
FPT	0.62	0.28	0.50	0.28		0.62	0.39	0.70	0.39
FPT_AMT	-	0.00	0.00	0.00		-	1.00	1.00	1.00
FPT_FLS	-	-	-	-		-	-	-	-
FPT_ITA	-	-	-	-		-	-	-	-
FPT_ITC	-	-	-	-		-	-	-	-
FPT_ITI	-	-	-	-		-	-	-	-
FPT_ITT	0.00	0.00	-	0.00		0.00	0.00	-	0.00
FPT_PHP	-	-	-	-		-	-	-	-
FPT_RCV	-	0.00	0.00	0.00		-	0.00	0.00	0.00
FPT_RPL	-	-	-	-		-	-	-	-
FPT_RVM	1.00	1.00	1.00	1.00		1.00	1.00	1.00	1.00
FPT_SEP	0.50	0.50	0.50	0.50		0.50	0.50	0.50	0.50
FPT_SSP	-	-	-	-		-	-	-	-
FPT_STM	1.00	1.00	1.00	1.00		1.00	1.00	1.00	1.00
FPT_TDC	-	0.00	-	0.00		-	0.00	-	0.00
FPT_TRC	-	0.00	-	0.00		-	0.00	-	0.00
FPT_TST	-	0.00	-	0.00		-	0.00	-	0.00
FRU	-	-	1.00	1.00		-	-	0.00	0.00
FRU_FLT	-	-	-	-		-	-	-	-
FRU_PRS	-	-	-	-		-	-	-	-
FRU_RSA	-	-	1.00	1.00		-	-	0.00	0.00
FTA	0.50	0.33	0.50	0.33		0.00	0.00	0.00	0.00
FTA_LSA	-	-	-	-		-	-	-	-
FTA_MCS	-	-	-	-		-	-	-	-
FTA_SSL	1.00	1.00	1.00	1.00		0.00	0.00	0.00	0.00
FTA_TAB	-	0.00	-	0.00		-	0.00	-	0.00
FTA_TAH	0.00	0.00	0.00	0.00		0.00	0.00	0.00	0.00
FTA_TSE	-	-	-	-		-	-	-	-
FTP	1.00	1.00	1.00	1.00		0.00	0.00	0.00	0.00
FTP_ITC	-	-	-	-		-	-	-	-
FTP_TRP	1.00	1.00	1.00	1.00		0.00	0.00	0.00	0.00

The reason why Linux, which is considered to be a more secure operating system (ComputerWire, 2004), receives lower total values in five out of 11 Security Classes (and only high values in two Classes) may be the fact that the evaluation is carried out without the influence of the environment; Microsoft Windows is the primary target for hackers and virus writers (ComputerWire, 2004) whereas Linux is more seldom attacked.

APPENDIX E
Heimdal Security Evaluator

An automation tool, by the name of Heimdal Security Evaluator, for generating evaluations, based on the Heimdal Framework, is presented here. The main window is illustrated in Figure 36. The software can be used for generating profiles for every step in the framework, and may also combine them into an evaluation report, which can be exported to various formats.

Heimdal Security Evaluator 3000 .NET contains five main tool windows; Evaluation Control, TOE Profile Manager, TOE Category Profile Manager, Reference Profile Manager, and Environment Profile Manager.

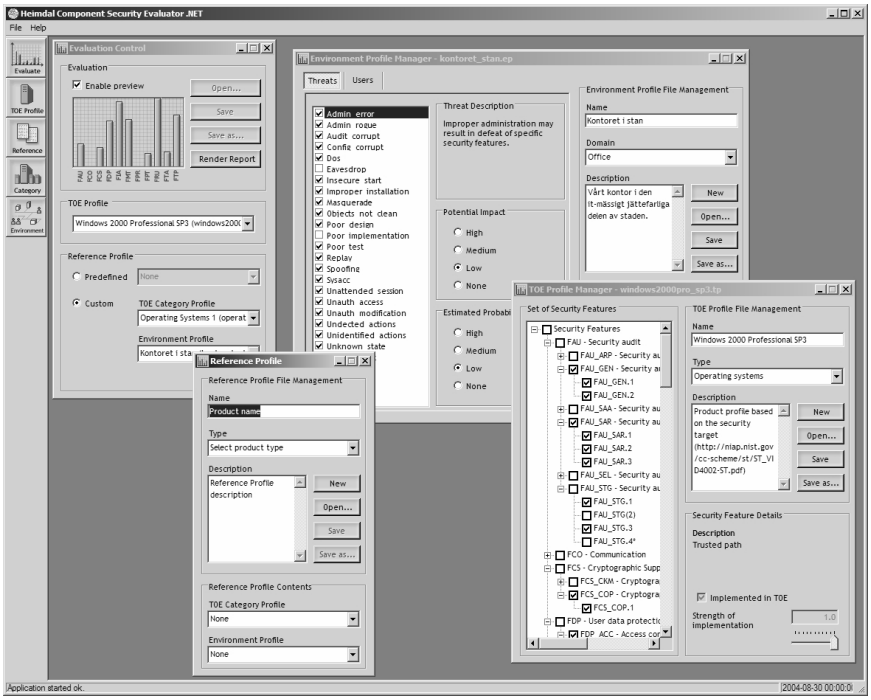


Figure 36: Main window of Security Evaluator 3000 .NET.

Evaluation Control

The Evaluation Control window (Figure 37) is used for combining a TOE profile, either with a predefined reference profile or with a TOE Category profile and/or an environment profile. The combo-boxes list all available profiles of the specific profile types.

A preview window displays the total security value for each of the 11 CC classes. The preview diagram is updated when a new profile is selected in either one of the combo-boxes, provided that the “Enable preview” checkbox is checked.

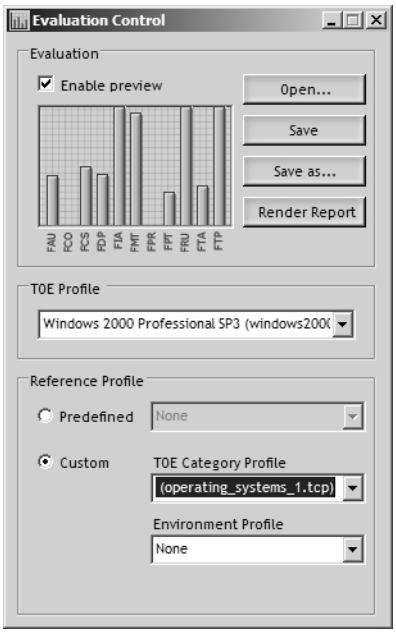


Figure 37: Evaluation Control window.

In order to get a more detailed picture of the security properties resulting from the evaluation, a user may render an evaluation report by clicking on the "Render Report" button.

TOE Profile Manager

The TOE Profile Manager window (Figure 38) is used to create a TOE Profile.

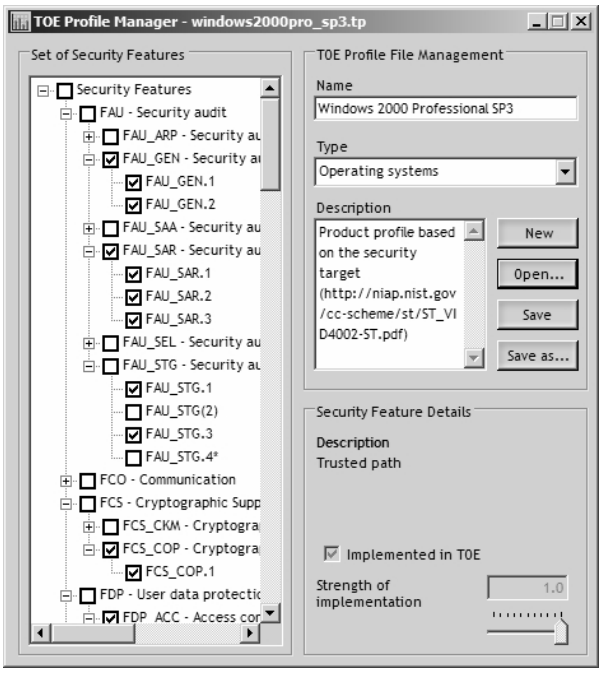


Figure 38: TOE Profile Manager.

The security features implemented in the TOE are set by checking the corresponding checkboxes in the Security Feature tree. When checking a node at the Class level the Group and Feature nodes in that class are checked. When a security feature is checked all its subsets are automatically checked as well.

A checked security feature is assigned the security value 1. The strength of implementation slider might, in future versions of the Evaluator 3000 .NET, be used to assign an arbitrary security value between 0 and 1 to the selected Security Feature. This is, however not implemented in the current version.

In order to use a TOE profile in an evaluation, it needs to be saved. This is handled by the TOE Profile File Management in the top-right corner.

TOE Category Profile Manager

The TOE Category Profile Manager window (Figure 39) is used to create a TOE Category Profile. The security features required by the category are set by checking the corresponding checkboxes in the Security Feature tree (in the left part of the manager window).

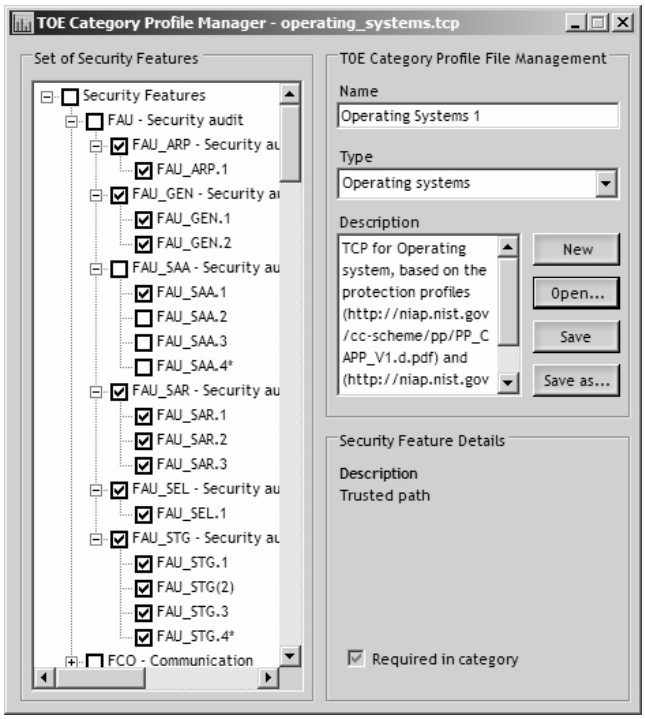


Figure 39: TOE Category Profile Manager.

When checking a node at the Class level the Group and Feature nodes in that class are checked. When a security feature is checked all its dependencies are automatically checked as well.

Reference Profile Manager

The Reference Profile Manager window (Figure 40) is used to combine previously saved TOE Category Profiles and Environment Profiles into a Reference Profile.

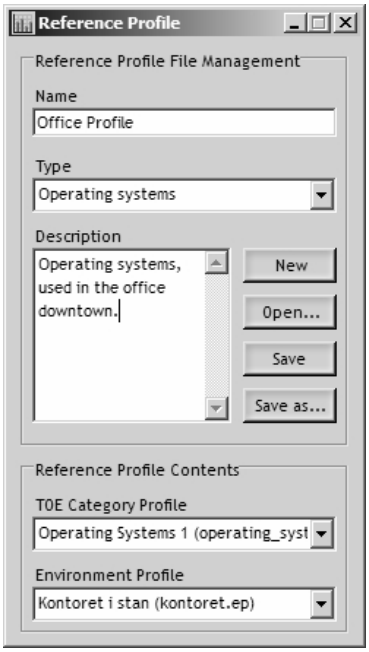


Figure 40: Reference Profile Manager.

Environment Profile Manager

The Environment Profile Manager window is used to create an Environment Profile, and consists of two tab pages; one is used to assign impact and probability values to threats (Figure 41), and one is the implementation of the user helper module (Figure 42).

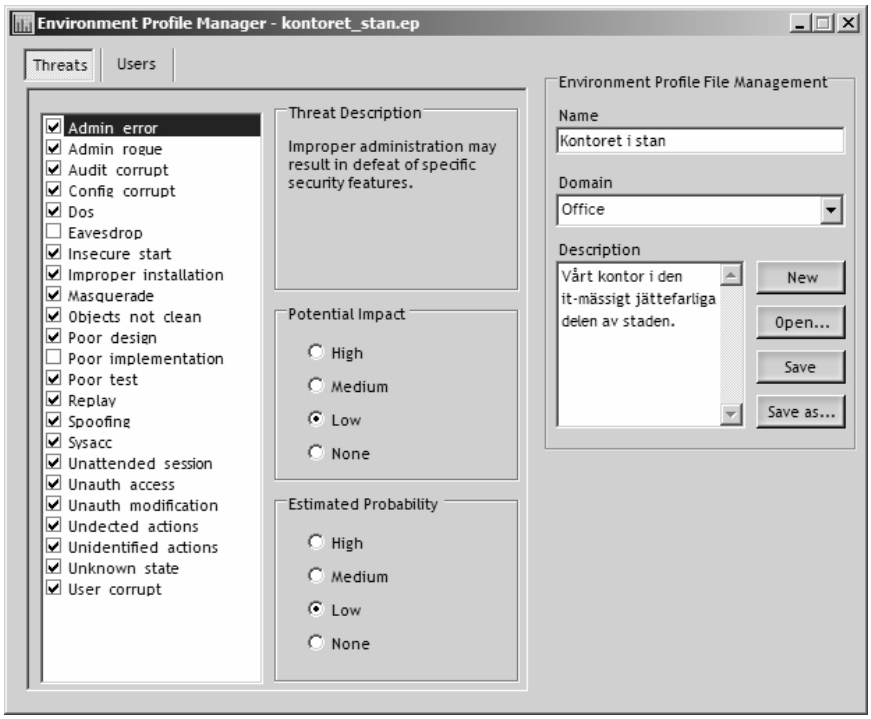


Figure 41: Environment Profile Manager – Threats.

For each threat in the threat list, the user determines the potential impact that threat would have on the organisation, as well as the estimated probability of the threat occurring.

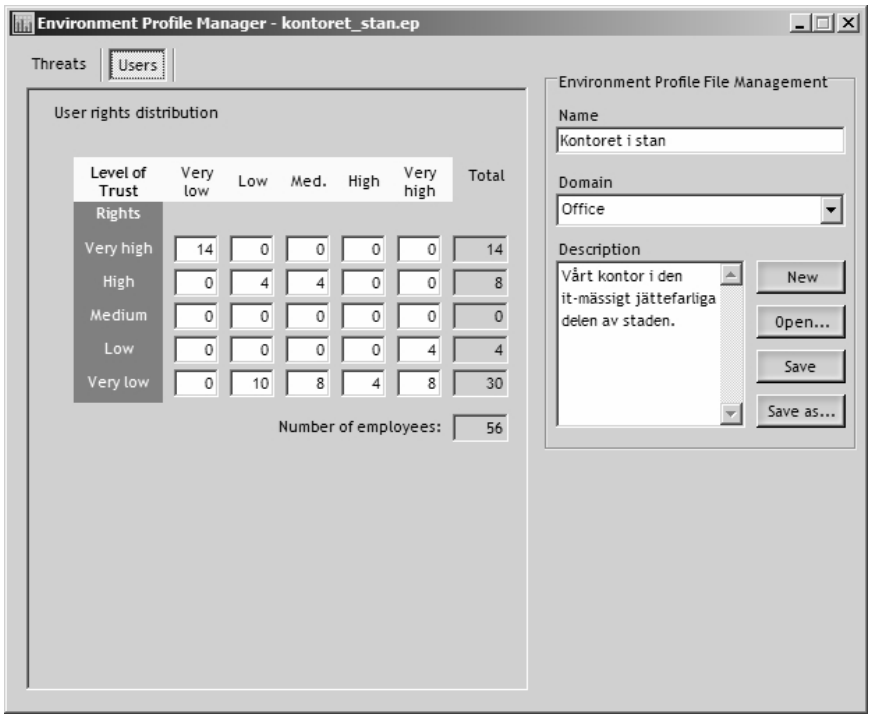


Figure 42: Environment Profile Manager – Users.

The user's trust-rights matrix is filled with the number of people in the organisation for each trust-rights box. A Risk Factor is computed based on this information, and combined with user-related threats from the threats tab.

Evaluation Report

The Evaluation Report window displays the results of an evaluation. The results may be displayed either as an overview (Figure 43), or in more details (Figure 44).

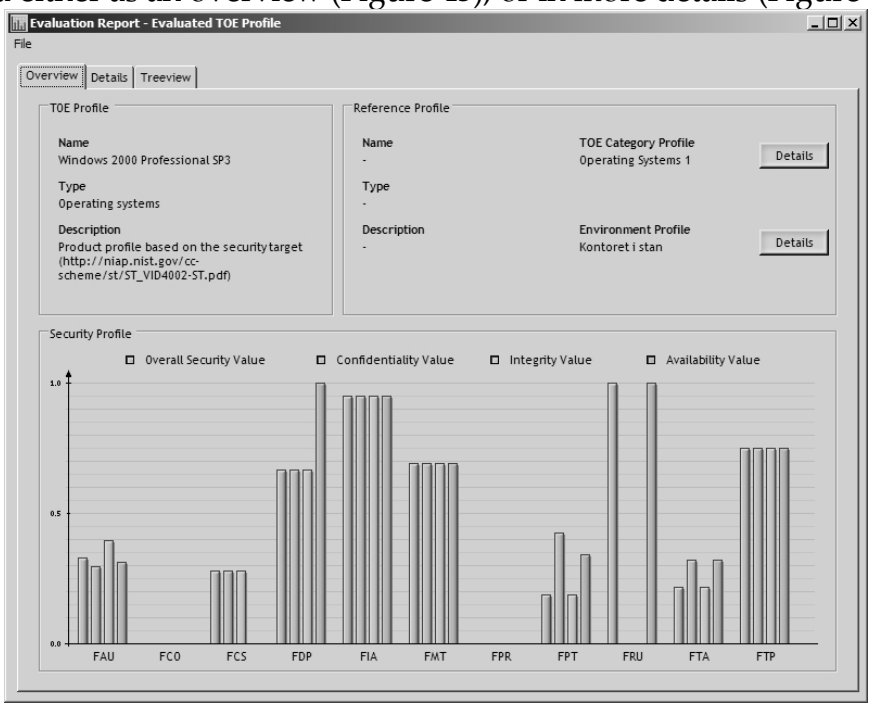
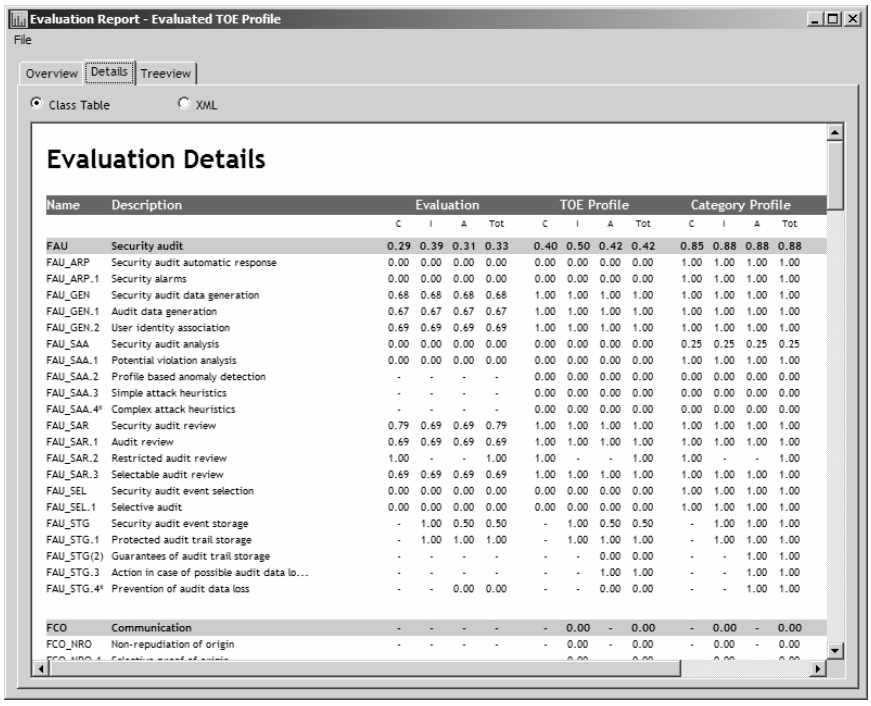


Figure 43: Evaluation Report – Overview.

The overview feature allows a user to view the evaluation on the Security Class level, including its CIA characteristics. Information about the profiles used in the evaluation is also accessible from the Overview tab.

The detailed view presents a user with the complete set of Security Values for the profiles used in the evaluation. The values are displayed down to Security Feature level, and enable a user to draw more detailed conclusions about weaknesses present in the TOE.



Name	Description	Evaluation				TOE Profile				Category Profile			
		C	I	A	Tot	C	I	A	Tot	C	I	A	Tot
FAU	**Security audit**	**0.29**	**0.39**	**0.31**	**0.33**	**0.40**	**0.50**	**0.42**	**0.42**	**0.85**	**0.88**	**0.88**	**0.88**
FAU_ARP	Security audit automatic response	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	1.00	1.00
FAU_ARP.1	Security alarms	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	1.00	1.00
FAU_GEN	Security audit data generation	0.68	0.68	0.68	0.68	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
FAU_GEN.1	Audit data generation	0.67	0.67	0.67	0.67	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
FAU_GEN.2	User identity association	0.69	0.69	0.69	0.69	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
FAU_SAA	Security audit analysis	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.25	0.25	0.25	0.25
FAU_SAA.1	Potential violation analysis	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	1.00	1.00
FAU_SAA.2	Profile based anomaly detection	-	-	-	-	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FAU_SAA.3	Simple attack heuristics	-	-	-	-	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FAU_SAA.4[a]	Complex attack heuristics	-	-	-	-	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FAU_SAR	Security audit review	0.79	0.69	0.69	0.79	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
FAU_SAR.1	Audit review	0.69	0.69	0.69	0.69	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
FAU_SAR.2	Restricted audit review	1.00	-	-	1.00	1.00	-	-	1.00	1.00	-	-	1.00
FAU_SAR.3	Selectable audit review	0.69	0.69	0.69	0.69	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
FAU_SEL	Security audit event selection	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	1.00	1.00
FAU_SEL.1	Selective audit	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	1.00	1.00
FAU_STG	Security audit event storage	-	1.00	0.50	0.50	-	1.00	0.50	0.50	-	1.00	1.00	1.00
FAU_STG.1	Protected audit trail storage	-	1.00	1.00	1.00	-	1.00	1.00	1.00	-	1.00	1.00	1.00
FAU_STG(2)	Guarantees of audit trail storage	-	-	-	-	-	-	0.00	0.00	-	-	1.00	1.00
FAU_STG.3	Action in case of possible audit data lo...	-	-	-	-	-	-	1.00	1.00	-	-	1.00	1.00
FAU_STG.4[a]	Prevention of audit data loss	-	-	0.00	0.00	-	-	0.00	0.00	-	-	1.00	1.00
FCO	**Communication**	**-**	**-**	**-**	**-**	**-**	**0.00**	**-**	**0.00**	**-**	**0.00**	**-**	**0.00**
FCO_NRO	Non-repudiation of origin	-	-	-	-	-	0.00	-	0.00	-	0.00	-	0.00

Figure 44: Evaluation Report – Details.

This table may be exported to tab-separated text, Microsoft Excel format, or to XML for use with other applications. The export functionality is accessed from the File menu.