ISO/IEC JTC 1/SC 27 **N6645**

ISO/IEC JTC 1/SC 27/WG 3 **N940**

REPLACES: N6297

ISO/IEC JTC 1/SC 27

Information technology – Security techniques

Secretariat: DIN, Germany

DOC. TYPE:	text for DTR ballot
TITLE:	Text for ISO/IEC DTR 15446:2008-06-25 -- Information technology -- Security techniques -- Guide for the production of Protection Profiles and Security Targets (JTC 1 N9154)
SOURCE:	Secretariat of JTC 1
DATE:	2008-06-25
PROJECT:	15446 (revision)
STATUS:	This document is currently undergoing a 2-month FDIS letter ballot at the JTC1 level. The P-members of JTC 1 and SC 27 are kindly requested to submit their votes on ISO/IEC DTR 15446 via the JTC 1 e-balloting application on by 2008-09-26. This document is being circulated within SC 27 for information.
ACTION:	**LB/FYI**
DUE DATE:	**2008-09-26**
DISTRIBUTION:	P-, O- and L-Members W. Fumy, SC 27 Chairman M. De Soete, SC 27 Vice-chair E. Humphreys, K. Naemura, M. Ohlin, M.-C. Kang, K. Rannenberg, WG-Conveners M. Nash, Project Editor
MEDIUM:	Livelink-server
NO. OF PAGES:	1 + 1- 89

ISO/IEC JTC 1/SC 27 N6645

ISO/IEC JTC 1 N 9154

2008-06-25

ISO/IEC JTC 1
Information Technology

Document Type:	**Text for DTR Ballot**
Document Title:	**ISO/IEC DTR 15446, "Information technology — Security techniques — Guide for the production of Protection Profiles and Security Targets"**
Document Source:	**SC 27 Secretariat**
Reference:	
Document Status:	**This document is circulated to JTC 1 National Bodies for DTR ballot. National Bodies are asked to vote and submit their comments via the on-line balloting system by the due date indicated.**
Action ID:	**VOTE**
Due Date:	**2008-09-26**
No. of Pages:	**89**

Reference number of working document: **ISO/IEC JTC 1/SC 27 N 6645**

Date: 2008-06-19

Reference number of document: **ISO/IEC DTR 15446**

Committee identification: ISO/IEC JTC 1/SC 27/WG 3

`Secretariat: DIN

Information technology — Security techniques — Guide for the production of Protection Profiles and Security Targets

Technologies de l'information — Techniques de sécurité — Guide pour la production de profils de protection et de cibles de sécurité

Document type: Technical Report
Document subtype: 3
Document stage: (50) Approval
Document language: E
C:\Data Files\TR 15446\DTR Revision\Text_DTR_15446.doc

Contents

Page

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

In exceptional circumstances, when a technical committee has collected data of a different kind from that which is normally published as an International Standard ("state of the art", for example), it may decide by a simple majority vote of its participating members to publish a Technical Report. A Technical Report is entirely informative in nature and does not have to be reviewed until the data it provides are considered to be no longer valid or useful.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO/IEC TR 15446 was prepared by Joint Technical Committee ISO/IEC JTC 1 Subcommittee 27 Security techniques.

This second edition cancels and replaces the first edition (ISO/IEC TR 15446:2004), which has been technically revised.

Introduction

This Technical Report is an adjunct to ISO/IEC 15408 Information technology – Security techniques -- Evaluation criteria for IT security. ISO/IEC 15408 introduces the concepts of *Protection Profiles* (PPs) and *Security Targets* (STs). A Protection Profile is an implementation-independent statement of security needs for a type of IT product that can then be evaluated against ISO/IEC 15408, whereas a Security Target is a statement of security needs for a specific ISO/IEC 15408 target of evaluation (TOE).

Unlike previous editions, ISO/IEC 15408:2008 provides a comprehensive explanation of *what* needs to go into a PP or ST. However, ISO/IEC 15408:2008 still does not provide any explanation or guidance of *how* to go about creating a PP or ST, or how to use a PP or ST in practice when specifying, designing or implementing secure systems.

This Technical Report is intended to fill that gap. It represents the collective experience over many years from leading experts in ISO/IEC 15408 evaluation and the development of secure IT products.

Information technology — Security techniques — Guide for the development of Protection Profiles and Security Targets

1 Scope

This document provides guidance relating to the construction of Protection Profiles (PPs) and Security Targets (STs) that are intended to be compliant with ISO/IEC 15408:2008. It is also applicable to PPs and STs compliant with Common Criteria Version 3.1 [1], a technically identical standard published by the Common Criteria Management Board, a consortium of governmental organisations involved in information security evaluation and certification.

As such, the document is primarily aimed at those who are involved in the development of PPs and STs. It will also be of interest to consumers and users of PPs and STs who wish to understand the contents of PPs and STs developed by others, and wish to confirm the relevance and accuracy of the information that they contain. It is also likely to be useful to evaluators of PPs and STs and to those who are responsible for monitoring PP and ST evaluation.

It is assumed that readers of this Technical Report are familiar with ISO/IEC 15408-1, and in particular Annexes A and B which describe STs and PPs respectively. PP and ST authors will (of course) need to become familiar with the other parts of ISO/IEC 15408 as described in this Report, including introductory material such as the functional requirements paradigm described in ISO/IEC 15408-2, clause 5.

This document is an informational ISO Technical Report intended for guidance only. It should not be cited as a Standard on the content or structure for the evaluation of PPs and STs. It is intended to be fully consistent with ISO/IEC 15408; however, in the event of any inconsistency between this Technical Report and ISO/IEC 15408, the latter as a normative Standard takes precedence.

This Technical Report does not deal with associated tasks beyond PP and ST specification such as PP registration and the handling of protected intellectual property.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 15408:2008 (all parts), *Information technology – Security techniques -- Evaluation criteria for IT security*

ISO/IEC 18045:2008, *Information technology – Security techniques -- Methodology for IT security evaluation*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 15408-1 apply.

4 Abbreviations

For the purposes of this document, the abbreviations given in ISO/IEC 15408-1 and the following apply.

COTS	Commercial Off The Shelf
SPD	Security Problem Definition

5 Purpose and structure of this technical report

This Technical Report is intended to help people who have to prepare Protection Profiles (PPs) or Security Targets (STs) for use in evaluation against International Standard ISO/IEC 15408:2008. It provides detailed guidance relating to the various parts of a PP or ST, and how they interrelate. This Technical Report is not intended as an introduction to evaluation using ISO/IEC 15408. Readers who seek such an introduction should read Part 1 of ISO/IEC 15408.

This Technical Report applies only to International Standard ISO/IEC 15408:2008. Earlier versions of ISO/IEC 15408 have different and incompatible technical requirements. However, the strategies proposed in this Technical Report will, in the main, be applicable to earlier versions of ISO/IEC 15408.

Clauses 1 to 4 contain introductory and reference material, and are followed by this overview clause (Clause 5).

Clause 6 provides an introduction to Protection Profiles and Security Targets – what they are, when and why they might be used. This clause also discusses the relationship between PPs and STs and issues relating to the PP/ST development process.

Clauses 7 to 13 provide information on how to specify the seven mandatory parts of the contents of a PP or ST, following the order outlined in ISO/IEC 15408-1, clauses A.2 and B.2.

Clause 14 examines the issues specific to PPs and STs for composed TOEs, i.e. TOEs that are composed of two or more component TOEs, each of which has its own PP or ST.

Clause 15 deals with some special cases, namely low assurance reduced PP/ST contents, conforming to national restrictions and interpretations and the use of functional and assurance packages.

Clause 16 discusses the topic of use of automated tools in PP/ST development.

6 An overview of PPs and STs

6.1 Introduction

This clause provides an overview of the roles of PPs and STs in information security evaluation using ISO/IEC 15408.

6.2 Audience

This document is intended for use by two distinct audiences:

a) IT professionals with security knowledge (e.g. security officers/architects with an understanding of a security requirement) but who are not experts in information security evaluation, and who have no prior knowledge of ISO/IEC 15408;

b) Experts in information security with good knowledge of ISO/IEC 15408, who are engaged in developing PPs and STs as part of their professional activities.

If you fall into the former category, this clause should provide you with the information you need to understand the purpose and structure of PPs and STs. It should also provide you with the background information you will need to read and understand PPs and STs, and to identify their relevance and correctness with respect to your particular circumstances. Following clauses will explain the contents of each part of PPs and STs in detail, but are oriented towards the production of such documents and assume knowledge of ISO/IEC 15408.

If you are an expert, you should already be familiar with the contents of this clause. Subsequent clauses will provide you with methodologies, techniques and practical tips that you can use to prepare PPs and STs in an efficient yet consistent manner.

If you are not an expert in information security, and you need to produce a PP or ST, this Technical Report will help you do so. However, you will also need to find, read and understand published examples of PPs or STs similar to your requirement. You should also consider calling on the services of others who *do* have the necessary specialist knowledge and experience.

6.3 The use of PPs and STs

6.3.1 Introduction

The main use of ISO/IEC 15408 is to assess the security of IT products. The term “IT product” is never actually defined in ISO/IEC 15408; however, it can be understood to cover any type of entity built using information technology, whether a complete IT system used exclusively by one organisation, or a COTS package created by a product manufacturer for sale to many different and unrelated customers. In this Technical Report, when we talk about IT products, or just products, our advice is intended to apply to all such entities. Where the scope of our advice is limited to a particular type of product, we talk about systems, or COTS products, or some other explicitly specific wording.

As IT products may be used in many ways, and in many types of environment, the notion of security will vary with the product. The end result of an ISO/IEC 15408 evaluation is therefore never “this IT product is secure”, but is always “this IT product meets this security specification”.

ISO/IEC 15408 has standardised security specifications to (among others):

- mandate specific content needed to assess a product against the security specification;

- allow comparison of security specifications of different products.

ISO/IEC 15408 recognises two different types of security specifications: Protection Profiles (PPs) and Security Targets (STs). The difference between these two is best explained by the roles they are intended to play in a typical product purchasing process, where a customer seeks to buy a product from a developer.

The notions of customer, developer and product are deliberately kept abstract. A customer is someone who wants to buy a product. It can be a single individual, an organization, a group of organizations, a government department etc. A developer is someone who wants to sell a product. It can be a single programmer, a small company, a large company, a group of companies working together etc. Finally, a product could be anything from a small software application or a smart card to a large operating system or a complete computer system containing hundreds of distinct components.

When our customer wishes to buy a product, he has essentially two possibilities:

- The customer contacts a developer, specifies his needs, and the developer creates a product that is specifically targeted towards that customer and exactly fulfils the demands of that customer. This may be expensive but the customer gets what he wants. In the remainder of this section, we will call this a specification-based purchasing process.

- The customer selects a product from a number of existing products. This is probably cheaper, but the resulting product may or may not exactly fulfil the customer's needs. In the remainder of this section we will call this a selection-based purchasing process.

When IT security is important, these purchasing processes have an added difficulty. For the average customer it is:

- hard to define what kind of IT security he needs;

- even harder to determine whether the IT security that a given product claims to have is useful or sufficient to meet his needs;

- and even harder to determine that if a product claims to have security properties, that these claims are true.

To assist a customer through a purchasing process and address the difficulties listed above, an evaluation of the product using ISO/IEC 15408 may be useful, and in this case, Protection Profiles and Security Targets play an important role. In the next two subsections, we will show how an evaluation may assist each type of process: specification-based and selection-based.

Of course, IT products do not work in isolation. The product is used by the customer in an operational environment, which may contain security measures of its own. Sometimes the product will make assumptions that certain types of security features exist within that operational environment. These assumptions will also form part of the PP or ST.

6.3.2 Specification-based purchasing processes

6.3.2.1 Overview

In a specification-based purchasing process, a customer writes a specification, provides this specification to a developer, and the developer then creates a product based on this specification. In more detail, the following steps must be performed:

a) The customer must determine his security requirements informally;

b) The customer must transform these informal security requirements into a more formal specification suitable for use by a developer;

c) The developer must build a product based on this specification.

In the end, the customer wants to know that "this product is useful for me". Therefore the quality of each of these steps is important.

6.3.2.2 Informal security requirements

The process of determining informal security requirements, that is determining "what is my security problem, and how should I address it?" is outside the scope of ISO/IEC 15408 and therefore outside the scope of this guide. However, this does not mean that this is unimportant or easy by any means.

Nevertheless, ISO/IEC 15408 assumes that the customer is capable of defining his or her informal security requirements. If this is done incorrectly, the product that is purchased in the end may not meet the true security requirements.

Customer requirements, once written down, often have a number of problems associated with them, especially in the area of security. Customer requirements are typically:

a) Incomplete (not all the requirements are present). For example, important threats that the product should counter are missing;

b) Not embedded: They are insufficiently tuned to the specific environment in which the product has to function, or do not describe this environment clearly enough.

c) Implicit: Some product requirements have consequences, but these consequences are themselves not included. The developer may not take these implicit requirements into account;

d) Not testable: The requirements are phrased ambiguously, so that it is not possible to verify whether a product meets the requirement or not;

e) Too detailed: The implementation has in fact already been written down but not the reason why this was chosen. If, in a later stage, the requirements change it is often unclear how these changes should be made;

f) Filled with ambiguous terms: like "the communication shall be secure" without defining what "secure" means;

g) Inconsistent: The requirements are internally self-contradictory.

Providing these customer requirements to a developer in a raw form will generally lead to problems, as the developer may misunderstand them. Security evaluation may lead to even more problems, since evaluators may interpret requirements differently from both the customer and the developer.

For these reasons, an important step in the whole specification-based purchasing process is the formalising of customer requirements. For security requirements based on ISO/IEC 15408, this formalisation takes place using a so-called Protection Profile (PP). A PP is in essence a document that defines the customer's security requirements in a formalised, standardised way.

6.3.2.3 Using PPs as specifications

PPs are typically written by large organizations, groups of organizations, government departments, etc. as they require a significant investment of effort.

A PP contains many sections, but as a security specification, the most important is the “security functional requirements”. Using ISO/IEC 15408, it is mandatory to write these requirements in a special language, defined within that International Standard. Use of this language ensures that the Protection Profile is:

a) not ambiguous: the language contains well defined terms, so that a developer can understand the requirements and interpret them correctly.

b) testable: the language is defined to contain only testable terms. Thus, it will be possible to assess in a later stage whether the product actually fulfils the PP.

c) not too detailed: the language enforces a certain level of abstraction. This closely follows what should be the consumer requirements: the consumer wants something to be done but does not want to worry how this is accomplished.

d) more complete: the language contains several constructions ("if this functionality is required then this other functionality is also required") to help ensure that implicit requirements are included.

6.3.2.4 Building a product from a PP

The customer can now give the PP, i.e. his formalised requirements, to a developer. The developer uses this PP as a starting point for the development of a product. As a first step in this process he writes a Security Target (ST).

An ST used for this purpose is very similar to a PP, but where a PP defines the customer requirements and is in principle written by the customer, the ST is a product specification and written by the developer.

The developer can of course not deliver an arbitrary ST as a reaction to the customer’s PP: his ST has to *conform* to the PP. This means that the product has to cover all the customer requirements, but:

- The ST may specify more than the PP: the product will offer more security functionality than the customer requirements (note: this extra functionality is not allowed to be incompatible with the PP), because, for example, the product will be sold to several customers, each with similar but slightly different requirements, or because the product is derived from an existing, standard product.

- The ST contains more detail than the PP: while the PP explains “what” shall be secured, the ST also explains “how”: the developer points out, in general terms, how he will implement the customer requirements.

A PP may permit the ST author flexibility to offer something that is equivalent but different in terms of security functionality provided – for more details, see 6.5.6.

The ST defines for the developer the security functionality his product should deliver and serves as a “Specification of Security Requirements” for the rest of the development process by the developer.

The result of the development process should be a product that can be delivered to the customer, who in turn can install it and use it. Naturally, this product should perform as described in the ST.

6.3.2.5 The role of evaluation in a specification-based purchasing process

Until now, we have only described the role of the customer and the role of the developer in this process. Based on this process, the developer could simply say to the customer (without further evidence):

a) my ST complies with your PP;

b) my product complies with my ST;

c) therefore, my product complies with your PP and meets your requirements.

If the customer accepts these statements, the process ends here.

However, if a customer requires independent verification of these statements, he can enlist a third party (an evaluation facility) to check these claims of compliance by performing an ISO/IEC 15408 security evaluation. In this process, an evaluation facility uses the PP, the ST, the product and ISO/IEC 15408 to assess two statements:

a) the ST complies with the PP;

b) the product complies with the ST.

Note that two issues are still left open, despite evaluation.

a) *The translation of the customer's informal security requirements to a Protection Profile.* As said earlier, this process falls outside the scope of ISO/IEC 15408, but if this is not done correctly, the PP will not match the customer's requirements and therefore the product will likely also not match the customer's requirements.

b) *Evaluation does not “prove” compliance*. An ISO/IEC 15408 evaluation will never provide an absolute guarantee that the product meets the PP, it can only deliver a certain degree of assurance depending on the depth and scope of evaluation as specified in the PP or ST.

6.3.3 Selection-based purchasing processes

6.3.3.1 Overview

The previous subsection discussed a customer delivering a specification and a developer implementing that specification. This subsection discusses the situation where a customer does not have the luxury of having a product made for him: he has to select from existing products. Therefore the purchase is no longer based on compliance to a formalised statement of customer requirements (i.e. a PP), but on comparison of existing products by the customer.

In a selection-based purchasing process of an IT product:

a) a developer must produce a product and a specification of this product and provide the specification to the customer.

b) the customer must determine from the specification (perhaps by comparing the specification to specifications from other developers) whether the specified product is the most suitable product for him to purchase.

As the customer in the end wants to know that “this product is suitable for me”, the quality of each of these steps is important.

6.3.3.2 Using a specification provided by the developer

In selection-based purchasing processes, the customer has to use a specification provided by the developer.

If this specification is informal, the same potential disadvantages hold as for the informal customer requirements discussed in 6.3.2.2. For this reason, this specification needs to be formalised as well. For this purpose ISO/IEC 15408 uses the Security Target (ST) as already discussed in 6.3.2.4. The ST here is identical to the ST discussed in 6.3.2.4, with one obvious difference: since it is not based on a customer’s PP, it cannot claim compliance to such a PP (it may claim compliance to other types of PP – see 6.3.4 below).

Because the developer does not know a specific customer’s requirements, he will have to make an estimate of what the market wants and codify this in the ST. This does therefore not necessarily match with any customer’s specific requirements.

The developer builds his product according to the ST: this process is similar to that described for specification-based purchasing processes.

6.3.3.3 Comparing Security Targets by the developer

The customer can now compare the STs of a number of products and select the one that best matches his requirements (probably also considering non-security requirements such as price). This means that he will still somehow have to find out what his informal security requirements are (see 6.3.2.2) and compare these with the STs offered to him. If one or more products match his requirements, he is done. If this is not the case, he will either have to choose the “closest” product or find some other solution (i.e. change his requirements).

As already stated in 6.3.2, the process of deriving informal customer security requirements falls outside the scope of ISO/IEC 15408 and this Technical Report. Comparing requirements and an ST also falls outside the scope of ISO/IEC 15408, although guidance on this topic will be found in later clauses of this Technical Report.

6.3.3.4 The role of evaluation in a selection-based purchasing process

Similar to the specification-based purchase process, the developer could simply claim that his product meets the ST and if the customer accepts this claim, the process ends here.

However, it is customary for the developer to offer a certificate confirming that an independent third party (an evaluation facility) has validated the ST, and then performed an ISO/IEC 15408 security evaluation to confirm that the product indeed meets the ST. It is even possible for the customer to commission the evaluation if he or she believes it to be essential and the developer has not done so.

Note that using evaluated products still leaves two issues open:

a) *Proving equivalence of the customer's informal security requirements and the Security Target.* As said earlier, this process falls outside the scope of ISO/IEC 15408, but if it is not done correctly, the ST may not match the customer's requirements, and therefore the product may not match the customer's requirements either.

b) *Evaluation does not "prove" compliance*. A ISO/IEC 15408 evaluation will never provide a perfect guarantee that the product meets the ST, it can only deliver a certain degree of assurance depending on the depth and scope of evaluation as specified in the ST.

6.3.4 Other uses of PPs

Protection Profiles have other uses. For example, standards bodies or vendor associations may specify PPs as best practice minimum security standards for specific types of applications. Governments and trade associations may mandate their use. Where these exist, both customers and developers are likely to require compliance with such PPs, as well as requiring or offering additional security functionality to meet their own specific needs.

Organisations specifying or mandating PPs for such purposes have an onerous responsibility to ensure that such PPs are minimal (they ask for no more than is absolutely necessary) and realistic (they do not ask for functionality or assurances that are not achievable by developers).

A PP may also be developed to express the need for a certain type of security product, even though it is recognised that at the time of publication, no such products (yet) exist. If you are a product developer, treat such PPs with caution. By the time you have developed a suitable product, the requirement may be obsolete or the sponsors of the PP may no longer want to buy your product because they have found other ways to meet their requirements.

Finally, PPs are *security* requirement specifications. Beware of their attempted misuse to specify other types of requirements which, if made more explicitly, would be rejected.

6.4 The PP/ST development process

The order of presentation of the requirements for PPs and STs in Annexes A and B of ISO/IEC 15408-1, and in earlier parts of this clause, might suggest that it is expected that PPs and STs are always developed in a logical 'top-down' manner, e.g. (in the case of an ST) that:

a) the security problem is first defined;

b) the security objectives are then identified to address the security problem;

c) security requirements are then defined to satisfy the security objectives for the TOE;

d) actual security functions are then selected to satisfy the security requirements.

Whilst such a possibility is not ruled out, it is more likely that an iterative process will be required. For example, definition of security requirements may highlight clarifications needed to the definition of the security objectives, or even the security problem. In general, a number of iterations may be required in which the relationships between threats, organisational security policies, security objectives and security requirements and functions are examined closely, particularly when rationales are being constructed. Only when all identified gaps in the rationales are filled may it be assumed that the PP or ST is complete.

During an iterative process of PP or ST development new information might surface, within the scope of the current security problem, that may lead changes to the document that reflect changes in external circumstances, for example:

a) new threats may be identified;

b) organisational security policies may change;

c) cost and time constraints may impose changes in division of responsibility between what the TOE is expected to do, and what is expected of the TOE environment;

d) changes in intended attack potential may impact on the TOE security problem definition.

It is also possible (particularly if the TOE is a product which has already been developed) that the PP or ST author already has a clear idea of the security functionality the TOE will provide (even if this has not yet been expressed as ISO/IEC 15408 security functional requirements). In such cases the definition of the security concerns and security objectives will unavoidably be influenced by the knowledge of the form of the security solution the TOE provides. The PP/ST development process will in those cases be, to some extent, ‘bottom-up’.

6.5 Reading and understanding PPs and STs

6.5.1 Introduction

This section is *not* intended for experts with prior knowledge of ISO/IEC 15408. It is intended for that part of the audience for this Technical Report that know very little about PPs or STs but who need to read one or more PPs or STs in order to understand the security capabilities of the related products. It is intended to highlight where potential omissions or deficiencies may be concealed, particularly in the scope of evaluation.

For detailed understanding of the contents of PPs and STs there is no substitute for reading ISO/IEC 15408-1, in particular Annexes A and B which provide details concerning Security Targets and Protection Profiles respectively. It is also a good idea to look at other PPs and STs that have been published and are in general use. There are a number of *registries* from which you can download these. The Common Criteria Portal includes the largest [2]. This register is recognised by the ISO and IEC councils as the official JTC 1 register for Protection Profiles and packages constructed in accordance with ISO/IEC 15408. It is operated in accordance with the relevant International Standard, ISO/IEC 15292 [3].

Unfortunately, a PP or ST cannot be summarised into a single number or a set of simple properties: PPs and STs describe a complex set of security properties that, if not carefully read, may lead to surprises when purchasing or using the product. On the other hand, some sections in a PP or ST (notably the security functional requirements) are equally or more important, but almost impossible to understand without an in-depth knowledge of ISO/IEC 15408. In the following subsections, we therefore identify the key sections of a PP or ST for the novice reader: sections that are relatively easy to understand, but that contain key information to understanding the security properties of a requirement expressed as a PP or a product described by an ST.

These relevant and readable sections are:

a) the TOE overview;

b) the TOE description;

c) the security objectives for the operational environment;

d) the conformance claim.

In the subsections following we will discuss each of these in more detail.

6.5.2 Reading the TOE overview

The TOE overview is in general the first thing you should read in a PP or ST, as it "is aimed at potential consumers of a TOE who are looking through lists of evaluated TOEs/Products to find TOEs that may meet their security needs, and are supported by their hardware, software and firmware" (ISO/IEC 15408-1, subclause A.4.2). The TOE overview contains three sections of interest:

a) Usage and major security features of the TOE;

b) The TOE type;

c) Required non-TOE hardware/software/firmware.

We will now discuss each of these in turn. You will find some simple examples of each in ISO/IEC 15408-1, subclause A.4.2.

The description of the usage and major security features of the TOE is intended to give a very general idea of what the TOE is capable of in terms of security, and what it can be used for in a security context.

This section should be fairly short (several paragraphs) so it should not require much effort to read and understand. And, as it should be aimed at consumers, it should not be highly technical. It is intended to be general, so it will not be exhaustive.

The TOE type is a description of the general category of IT products the TOE belongs to (e.g. firewall, smart cards, intranet, LAN etc.). ISO/IEC 15408 mandates that the TOE overview lists any reasonable expectations that a reader may have from this TOE type but that are not supported by the TOE. Specifically:

a) If the TOE-type would lead you to believe that the TOE has certain security functionality and it does not have this functionality, the TOE overview must list this missing functionality.

b) If the TOE-type would lead you to believe that the TOE could be used in a certain environment and it can not be used in such an environment, the TOE overview must list this.

Note that this is the only place in a PP or ST where these warnings are required to appear. The writer of the PP or ST may repeat this information in appropriate places later on by means of notes, but is not required to do so.

If these warnings are provided and possibly impact upon your intended use, you should seriously consider whether you can still use this TOE with these limitations.

The TOE, especially when it is a software type TOE, will sometimes have to rely on hardware and possibly firmware and other software components just to be able to execute. If this is the case, the TOE overview is required to identify this non-TOE hardware/software/firmware.

The PP or ST does not have to provide a complete and fully detailed identification of all this hardware/software/firmware, but the identification should be complete and detailed enough for you to determine the major external hardware/software/firmware components needed to use the TOE.

You should carefully assess whether there are any non-standard components on which the TOE relies and whether these components fit in with your existing infrastructure, budgets, company policies etc.

6.5.3 Reading the TOE description

An important thing to understand about ISO/IEC 15408 evaluations, is that if you read that well-known product XYZ has been evaluated, this does not mean that all security features (or even a majority of security features) of this product have been evaluated. It may well be the case that only some of its security functional features have actually been looked at and the remaining ones were not considered part of the evaluated security functionality. ISO/IEC 15408-1, subclause A.4.1 prohibits misleading TOE references, but developers can always get around this by just using a product name. You need to check that the evaluated functionality meets your needs. If some of the security functionality you intend to use was excluded, you need to ask yourself why.

One of the most important roles of the TOE description is to allow the ST reader to find this out. To this end the TOE description discusses the physical and logical scope of the TOE in detail.

Starting with the physical scope, ISO/IEC 15408 tells us that “The TOE description discusses the physical scope of the TOE: a list of all hardware, firmware, software and guidance parts that constitute the TOE. This list should be described at a level of detail that is sufficient to give the reader a general understanding of those parts” (quotation from ISO/IEC 15408-1, subclause A.4.3).

You should briefly examine this list to see if you see anything odd in it that you would not expect, or whether some parts of the product that you might expect to find present are missing. If something is not in this list, then the evaluation has completely ignored it and assumed it did not exist. If you intend to use that part, then you can draw no conclusions about its security capabilities from the evaluation.

With regards to logical scope, ISO/IEC 15408 tells us that “The TOE description should also discuss the logical scope of the TOE: the logical security features offered by the TOE at a level of detail that is sufficient to give the reader a general understanding of those features. This description is expected to be in more detail than the major security features described in the TOE overview” (quotation from ISO/IEC 15408-1, subclause A.4.3).

Whereas the physical scope tells us the list of parts of the TOE, the logical scope should tell us what the TOE does. This was already briefly discussed in the Usage and Major Security features section (see 6.5.2) but where that discussion was only a few paragraphs, this discussion is more likely to be a few pages. The most important feature of this section is that if you expect the product to have a certain feature such as remote management (e.g. because an advertisement of the product in a trade magazine describes that feature) but the logical scope does not mention remote management, it may well be that remote management was not evaluated, and hence, remote management should not be turned on if you want to use the product in its evaluated configuration.

It is therefore important to scrutinise this section to determine whether all security-related features that you require were actually evaluated. If they are not, you will have no assurance in the operation of that feature from evaluation.

6.5.4 Security objectives for the operational environment

The operational environment is the general location that the TOE will be placed in. In order for the TOE to work correctly, this operational environment must meet certain constraints. For example, if a TOE is a high-availability server, this TOE needs to be protected against people accessing it with a screwdriver. This protection could be provided by the TOE (although tamper-proof servers are pretty rare), so in general the operational environment should address this, by specifying a requirement for a locked secure server room.

These and similar requirements for the operational environment are described in a PP or ST in the security objectives for the operational environment section. These objectives describe the things that must be achieved by everything except the TOE in order for the TOE to meet its security requirements. You will find a number of example of security objectives for the operational environment in ISO/IEC 15408-1, subclause A.7.2.2.

It is of vital importance to realize that these are not guidelines, but necessary conditions for the TOE to operate as stated. All of these objectives must be fully met and addressed by you or your organisation: the TOE will not do it for you. If a single one of these objectives is not met, the TOE might not function securely. It is therefore imperative that you determine whether they are achievable in your organisation, and if one of them is not achievable, the TOE may not be suitable for you.

6.5.5 Reading the conformance claim

The conformance claim is usually found in a prominent place in the PP or ST, usually somewhere up front. It usually consists of a single sentence of the form:

This Protection Profile/Security Target claims conformance to:

- ISO/IEC 15408:2008. This part of the claim represents the version of ISO/IEC 15408 that is used. If this is not 2008 or higher (or the Common Criteria equivalent V3.1 or higher), the PP/ST will not match the specifications in this Technical Report, and this Technical Report is not directly applicable.
- Part 2 extended or Part 2 conformant. This part of the claim defines how security functional requirements are constructed, and from a consumer point of view, both are acceptable.
- Part 3 extended or Part 3 conformant. This part of the claim defines how security assurance requirements are constructed. If the answer is “Part 3 extended”, the developer of the PP and ST has designed their own assurance tests, and from a consumer point of view, you should question why this was necessary.
- a list of packages that the TOE claims conformance with. Usually there is only one such package and it is named EAL1, EAL2, ..., EAL7, often followed by “augmented”. These EALs are discussed further in 6.5.7. It is rare to find any other type of package specified and from a consumer point of view, you should again question why this was necessary.
- a list of Protection Profiles that the PP or ST claims conformance with. This is discussed further in 6.5.6 below.

6.5.6 Conformance to Protection Profiles

As already described in 6.3.2.4, STs can claim conformance to PPs (but do not have to do so). Also, PPs can claim conformance to other PPs. If they do claim conformance, this is listed here. ISO/IEC 15408 does not allow any form of partial conformance, so if a PP is listed here, the PP or ST must fully conform to the referenced PP or PPs.

Conformance to a PP means that the PP or ST (and if an ST is of an evaluated product, the product as well) meets all requirements of that PP.

If you are reading a PP, you will also find a statement that STs and other PPs must conform in a way that is either “strict conformance” or “demonstrable conformance”. Published PPs will normally require demonstrable conformance. This means that STs claiming conformance with the PP must offer a solution to the generic security problem described in the PP, but can do so in any way that is equivalent or more restrictive to that described in the PP. “Equivalent but more restrictive” is defined at length within ISO/IEC 15408, but in principle it means that the PP and ST may contain entirely different statements that discuss different entities, use different concepts etc., provided that overall the ST levies the same or more restrictions on the TOE, and the same or less restrictions on the operational environment of the TOE.

Strict conformance is only used where no differences are permitted between PP and ST, e.g. in selection based purchasing (see 6.3.3 above). Of course, an ST can still introduce additional restrictions if it wishes to do so. If a PP demands strict conformance, and you or your organisation did not write it, it is highly unlikely to be suitable for your use.

6.5.7 EALs and other assurance issues

The TOE overview and TOE description will tell you what the TOE is capable of doing, i.e. the functionality that is provided by the TOE. However, functionality does not say everything about a IT product. Products with the same general functionality can be used in different settings. For example, the same smart card design can be used as:

- a bus ticket with a small amount of "travel budget" on it;
- a credit card with a €10.000 credit limit;
- an access control measure for access to a top secret military facility.

In the first case, one is happy with a "low-quality" smart card. If a hacker manages to break the bus ticket, he may be able to get free bus rides until the card parameters change. The loss of potential revenue (provided that other cards are not hacked in the same way) is not significant to the bus company.

In the second case, and certainly in the third case, we need much more confidence in the correct implementation of the card functionality, as the consequences of breaking even one card may be significant.

In ISO/IEC 15408, this quality is called "assurance". ISO/IEC 15408 measures assurance by examining many aspects of the development of the product, such as the development and production process, the designs, the manuals, the amount of testing done by the developer of the product etc.

ISO/IEC 15408 formalises assurance into 27 categories (the so-called assurance families). In each category, ISO/IEC 15408 specifies different levels of conformance, where meeting a higher level is better.

As an example, a product could score in the category developer test coverage:

- 0: it is not known whether the developer has performed tests on the product;
- 1: the developer has performed some tests on some interfaces of the product;
- 2: the developer has performed some tests on all interfaces of the product;
- 3: the developer has performed a very large amount of tests on all interfaces of the product.

It can be seen from that this example that the degree of effort expended increases with each level, and the degree of uncertainty decreases.

Unfortunately it is almost impossible for a non-expert to interpret a scorecard consisting of individual ratings for all 27 subcategories. To allow non-experts to assess assurance, ISO/IEC 15408 has 7 predefined ratings, called Evaluation Assurance Levels (EALs). These are called EAL 1 to EAL 7, with EAL 1 the lowest and EAL 7 the highest.

Each EAL can be thought of as a set of 27 numbers, one for each subcategory. For instance, EAL1 assigns a rating of 1 to 13 of the subcategories, and 0 to the other 14 subcategories, while EAL2 assigns the rating 2 to 7 subcategories, the rating 1 to 11 subcategories, and 0 to the other 9.

The EALs are strictly hierarchical, so that if EAL n assigns a certain rating to a certain subcategory, then EAL n+1 will assign the same or a higher rating to that subcategory. So EAL n+1 provides strictly more assurance than EAL n overall.

The drawback of higher assurance is of course cost. In the test coverage area described earlier, a rating of 0 will mean no cost, but for each higher rating, the developer will have to perform and document the tests that are being done, the evaluator will have to determine if the developer did this correctly and document this, etc. More assurance almost always means more cost. Of course, more assurance also reduces the risk that the claimed functionality does not work correctly or contains exploitable vulnerabilities.

A listing of each EAL, together with a description of that EAL and a characterisation of the assurance that that EAL provides can be found in ISO/IEC 15408-3, clause 8.

EALs are a broad-brush mechanism, and are more suitable for assessing some types of product than others. Nevertheless, the EALs are currently the only widely accepted way to provide a characterisation of ISO/IEC 15408 assurance that a relative layman can understand.

6.5.8 Summary

In summary, this section was intended to convey two things:

a) (obviously) that an ST can be reasonably understood from reading a number of sections; but also

b) (less obviously) that these sections may contain important caveats and are therefore vital to understanding the limitations of the evaluation.

In the past there have been cases where consumers have stated that they wanted an EAL4 firewall or whatever. Hopefully, this section has conveyed that a ISO/IEC 15408-certified EAL4 firewall may have limitations that make it totally unusable for you, and may not provide all the relevant security you need.

For example, suppose you need both packet routing and HTTP/FTP proxy services from your firewall. A router may have a TOE type of firewall, and have been evaluated at EAL4. But as a router, it will only offer packet routing controls. Worse, if you find an evaluated firewall that offers proxy services but the logical scope is limited to packet routing, you must ask yourself why.

Even a big standard like ISO/IEC 15408 is not a substitute for thinking, and complex matters like IT security cannot be reduced to one sentence descriptions, no matter how hard you try.

6.5.9 Further reading

The sections of the PP or ST described above are the most basic sections of the PP and ST, and the most useful to be read by relative laymen. If you want to know more about the product, you could also try reading the TOE summary specification, which is intended to provide more detail on how the TOE is implemented. This section does not have to be easily readable. It may be filled with unexplained abbreviations like FIA_UID.2.1. However, many developers will take great pride in producing a TOE summary specification that meets the requirements of the evaluators but can still be readily understood by users of the product.

If you need to understand other sections of a PP or ST, then the following clauses of this Technical Report may help you. Whilst they are designed to be used by experts to specify PPs and STs, the information should also help you to understand the relevant contents.

7 Specifying the PP/ST introduction

This clause provides guidance on the specification of the PP/ST introduction section of a PP or ST. These are described at length in ISO/IEC 15408-1, clauses A.4 and B.4, and therefore little additional guidance is necessary in this Technical Report.

The introduction of a PP consists of the following elements:

- a PP reference;
- a TOE overview.

The introduction of an ST consists of the following elements:

- ST and TOE references
- a TOE overview;
- a TOE description.

The only non-obvious part is the “usage and major security features of the TOE” section of the TOE overview. Usage is often best derived by summarising the security problem definition section of the PP or ST (see clause 9 for details), whilst the major security features are best described by summarising the security objectives for the TOE. This will ensure that the introduction is consistent with the more detailed parts of the PP or ST.

As with most introductions, you will probably find it easiest if you leave it until the rest of the PP or ST is complete and write it last!

8 Specifying conformance claims

This clause provides guidance on the specification of the conformance claims section of a PP or ST. ST conformance claims are described in ISO/IEC 15408-1, clause A.5, and the differences applicable to conformance claims in PP in ISO/IEC 15408-1, clause B.5.

The Conformance Claims section of a PP or ST describes how the PP or ST conforms to:

a) *ISO/IEC 15408.* This consists of listing the exact version of ISO/IEC 15408 that was used to write (and presumably also to evaluate) the PP or ST. If an unofficial translation of ISO/IEC 15408 into some language other than English was used, this should also be indicated. If any corrigenda, or CC interpretations or supporting documents were used, these should be listed as well.

b) *Protection Profiles.* This consists of listing any Protection Profiles that this PP or ST claims conformance to. A simple list is sufficient: no extra information is needed in this section.

c) *Packages.* This consists of listing any packages that are referenced by the PP or ST. It is normal to claim conformance to one of the assurance packages (EALs) defined in ISO/IEC 15408-3, possibly with augmentations. Use of packages is discussed further in 15.3. Again: a simple list suffices: no extra information is need in this section.

Of course, this conformance also applies to any TOE based on that PP or ST.

If you are specifying a PP, you must define how other PPs and STs conform to your PP. There are two choices for this:

a) Strict. Conceptually, this means that conforming PPs/STs must contain everything in this PP. See ISO/IEC 15408-1, subclause 8.3 for the precise requirements.

b) Demonstrable. Conceptually, this means that conforming PPs/STs must be “equivalent” to this PP. Again, please see ISO/IEC 15408-1, subclause 8.3 for the precise requirements.

As a guideline, if you are writing a PP as the precise and complete specification for a product you will be buying or building for your own private use, you should require “strict” conformance. If you are specifying a PP for any other purpose, use “demonstrable”.

If you claim conformance to a functional package or another PP, your own security problem definition, security objectives and security requirements must be compatible with that package or PP.

If you are writing a PP or ST and add additional requirements to those found in a referenced PP, be very careful that you do not create inconsistencies such that no TOE can implement all of the requirements, all at the same time.

9 Specifying the security problem definition

9.1 Introduction

This clause provides guidance on the specification of the *security problem definition* (SPD) section of a PP or ST. ISO/IEC 15408-1, clauses A.6 and B.6 describe PP and ST SPDs respectively. ISO/IEC 15408-1, clause B.6, which deals with PPs, is simply a pointer to A.6, which can be taken as a confirmation that the expected content of the security problem definition section does not differ between a PP and an ST. Indeed, the wording of the relevant validation criteria in ISO/IEC 15408-3 is identical.

The purpose of the security problem definition is to define in a formal manner the nature and scope of the security problem which the TOE is intended to address. This is illustrated in Figure 1 following.

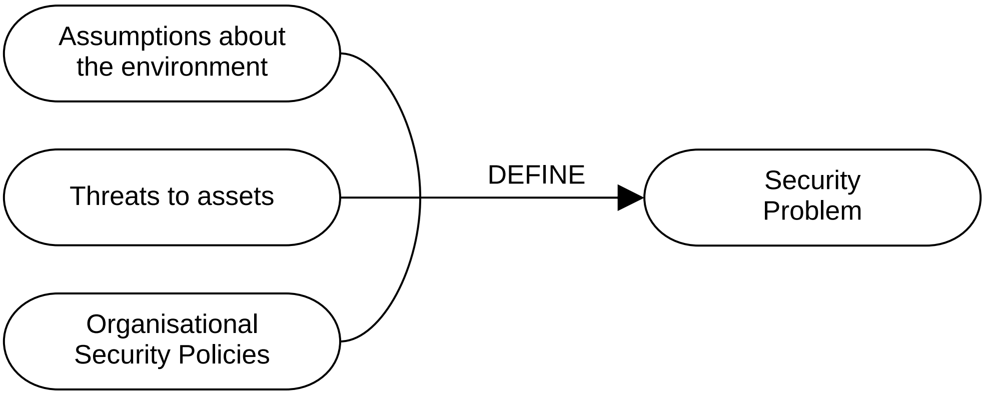
Assumptions about the environment
Threats to assets
Organisational Security Policies
DEFINE
Security Problem

Figure 1 — Security problem definition

Although not all Protection Profiles and Security Targets contain a security problem definition (see clause 15), where it is present it is probably the most important part of the PP or ST, and the most dangerous to delegate to external contractors to prepare. To quote from ISO/IEC 15408:

“The usefulness of the results of an evaluation strongly depends on the ST, and the usefulness of the ST strongly depends on the quality of the security problem definition. It is therefore often worthwhile to spend significant resources and use well-defined processes and analyses to derive a good security problem definition”, (ISO/IEC 15408-1, subclause A.6.1).

If the problem specified is the wrong problem, or if it is ambiguously described, then the remainder of the PP or ST will also be wrong. Worse, the wrong product may be selected or purchased on the basis of a valid but inapplicable specification. This clause is therefore one of the largest and most detailed in this Technical

Report, although the criteria that it describes in ISO/IEC 15408 occupy only two or three pages of text. Regardless of whether you are a developer or a customer, and regardless of whether your PP or ST will be used in a specification or selection based process, it is important *to get the security problem definition right*.

Subsequent sections of the PP and ST show how the security problem will be addressed by the TOE, in combination with its operating environment. It is therefore important to ensure that the security problem definition is clear, concise and consistent.

ISO/IEC 15408 does not assume or mandate any particular process or methodology for preparing the security problem definition; you can use any method you like. Of course, if you are new to the process of developing PPs and STs this is not helpful. This clause therefore includes a detailed description of a simple methodology that has been tried and tested in practice and found to work in a variety of organisations and environments. It is based upon a series of steps, performed in sequence:

a) Identifying and confirming the informal security requirement;

b) Identifying and specifying the applicable threats by performing formal threat analysis;

c) Documenting the applicable policies;

d) Documenting the applicable assumptions;

e) Finalising and checking the complete SPD specification.

Regardless of the methodology employed, this Technical Report assumes that the security problem definition represents a formalised description of an existing informal security requirement. Of course, in practice, there may not be a straightforward single document that represents that informal requirement, it may not even be written down! The first step in the recommended methodology is therefore to identify and confirm the informal requirement, even though it does not appear within the PP or ST. The informal requirement may be obvious and well defined. In other cases, a large part of the work in developing the SPD may simply be identifying the informal requirement, and obtaining confirmation from management and other stakeholders that it is a correct and complete representation of their security needs.

The methodology also has two other aspects that are not *required* by ISO/IEC 15408, but which have been found in practice to save time overall, by avoiding confusion and queries in later stages of PP/ST development. These are:

a) Documenting discounted threats;

b) Producing a rationale to link the SPD back to the informal security requirement.

Both of these are explained in more detail at appropriate points in the methodology, but in brief, discounted threats are threats that may or may not apply to the product, but which, if applicable, would be countered by security functionality included in the TOE for other reasons. If these are not documented in the SPD, they are likely to be raised as queries when the PP/ST is reviewed. More seriously, if the requirement changes, functionality might be removed without considering its value in also covering discounted threats.

Evaluation treats the SPD as axiomatic; no attempt is made to trace it back to actual security needs. If no SPD rationale is created, there is always a risk that parts of the informal requirement may be lost in the process of creating the SPD, and that this is not discovered until the product is used and found not to be fit for purpose. A rationale therefore provides an important consistency and completeness check.

As a general principle, the security problem definition should avoid, where possible, any discussion of the form of the TOE's response to meeting the its requirements, e.g. details relating to the TOE security functions. By following this principle, you will help to focus the reader's attention on what are the important aspects of the security problem. Discussion of how the security problem will be satisfied by the TOE should be left to the later parts of the PP or ST. Of course, where a particular solution is mandated as part of the informal security requirement, that solution will have to be stated as part of the SPD, both to ensure it is documented and as justification for constraining later design decisions.

9.2 Identifying the informal security requirement

9.2.1 Introduction

There are always many things about a security problem – and its intended solution – that are already fixed and known before security problem definition begins. These requirements and constraints form the informal security requirement. The difficulty is always to identify and document them. This therefore becomes the first step in our recommended methodology.

9.2.2 Sources of information

9.2.2.1 Overview

There are many ways that aspects of the informal security requirement can be identified. The subsections following discuss some of them. In a particular organisation, there may be others that a generic methodology as described in this Technical Report cannot identify. You will have to think about your security needs carefully and thoroughly. However, the possible sources of information suggested in this subsection should help.

9.2.2.2 Required functionality

Security functionality may be part of the purpose of the product under consideration. This particularly applies to COTS products, where security services to be available to the purchaser through Application Program Interfaces (APIs) or Human Computer Interfaces (HCIs) may be an essential part of the product specification.

If security functionality is part of a documented user requirement, providing it is part of the problem addressed in the SPD.

9.2.2.3 Risk assessment

A security risk assessment may have already been performed covering a proposed system, and even a COTS product, and identified risks that need to be reduced by IT security controls. These risks represent part of the security problem.

There are many methodologies for performing risk assessments. However, these methodologies generally accept that for a risk to exist, there must be three things: an asset with a value that can be damaged in some way, a threat, something or someone who can damage the asset, and a vulnerability, a way that the asset can be damaged. If any one of these three does not exist, there can be no risk. This form of model is assumed by ISO/IEC 15408; if the actual risk assessment used an incompatible model of risk, there might be problems mapping its results into a suitable form for use in the SPD.

9.2.2.4 Threat assessment

A threat assessment is a weakened form of risk assessment where it is assumed that if a threat exists, assets can be damaged and thus a risk will exist. In this case, the identified threats represent part of the security problem.

Threat assessment is particularly appropriate where the person trying to identify and specify a security problem is not the owner of the assets that will be protected, and thus not in a position to perform risk assessment or determine the value of assets.

9.2.2.5 Management policy

A security requirement can result from a policy decision by management, for example that all systems in a particular organisation will contain certain standard IT security controls. This process is sometimes known as "minimum standards" or "risk avoidance". The policy may be arbitrary, for example, following what similar organisations do, or it may have a logical basis, for example to meet legal requirements or contractual conditions imposed by customers.

Of course, even where a policy has a logical basis in law or contract, the mandated security controls may not be appropriate for a particular system or organisation, or may only be applicable in part.

9.2.2.6 Presentational policy

A security requirement may arise from a wish to demonstrate that an organisation or a COTS product implements certain IT security controls. This policy may arise due to marketing needs, or from a wish to be seen to follow best practice.

Security problems of this type are well suited to ISO/IEC 15408 evaluation, as successful evaluation using an approved evaluation facility will permit an official certificate to be issued, providing independent proof that the controls exist. Published PPs can be used to identify suitable controls.

The drawback to policies of this type is that they are based on achieving certification or demonstration of compliance, not in selecting security controls that are relevant to the product in question. This can cause problems finding reasons to put in the SPD that justify the need for the controls. They may have to be treated as policy decisions, which the originator may be reluctant to acknowledge is the true reason for their selection.

9.2.2.7 Evaluation policy

An organisation may have a policy that IT products are evaluated using an evaluation scheme based on ISO/IEC 15408 or the Common Criteria, regardless of the security controls they implement.

This requirement is problematic. The security problem to be addressed forms no part of the policy and is therefore not properly defined. However, such policies are found in practice, and do result in requirements for STs to be prepared.

9.2.3 Documenting the informal requirement

The best source of information about a security problem is the results from a security risk assessment. If you are lucky enough to have access to the results from a risk assessment, not only is it likely to be comprehensive, but most risk assessment methodologies introduce the concept of proportionality, where risks can be tolerated, so long as the likelihood of a loss is very low or the consequences of a loss are not significant. Identifying both acceptable and unacceptable risks enables the security problem to be modified later through design trades. If the controls required to eliminate particular risks turn out to be difficult to implement or difficult to evaluate, an acceptable overall level of risk can still be achieved by using different controls in different ways to counter different potential risks.

Of course, a risk assessment prepared by a third party for their own purposes may not judge risk in the same way that you would do. In such cases, use their results with caution.

If describing part of the problem in terms of risks is not possible, it is almost certain to have an arbitrary basis that cannot be modified or amended. It is important that this is made clear in the informal description.

Relevant information may relate not only to the IT product to be developed, but also to its operating environment. The operating environment determines the level of reliance that can be placed on personnel, procedural and physical controls. A public space is very different in its security needs to a locked server room. If it has been established that certain personnel, procedural and physical controls can be assumed to be in place, that will be an important part of the security problem definition.

As well as information about risks and controls, design decisions may have already been made about how certain security functions are to be implemented – for example, a decision to use biometric authentication rather than passwords, or to use certain communications protocols such as http/https that have defined security characteristics.

Some parts of a security problem may not be solvable by technical means; they may only be countered by personnel, procedural and physical controls. They are still part of the security problem, and need to be documented. Indeed, any aspect of the security problem that has already been decided should be documented as part of the informal security requirement.

When all the information available has been identified, collated and checked for inconsistencies, it should then be divided into three areas:

a) potential attacks that the product must counter;

b) security attributes or features that the product must possess; and

c) security attributes or features that the product need not possess.

These distinctions are important, as they are treated in subsequent steps in different ways. Potential attacks must be treated as threats to the TOE and countered. Security attributes and features that the product must possess, including mandated security solutions, correspond to organisational security policies (OSPs). Attributes and features that the product need not possess correspond to assumptions. We deal with each of these in turn in subsequent subclauses.

Different parts of the informal requirement derived from different sources may overlap or may even be inconsistent. It is not uncommon for security attributes or features to be mandated as a subconscious response to identified potential attacks. Similarly, certain types of attack may be subconsciously considered too difficult or too expensive to counter effectively, and so relevant security features declared as not necessary. Such inconsistencies need to be sorted out before the informal specification is taken any further. Your aim should be to express each aspect of the informal requirement once and once only.

9.3 How to identify and specify threats

9.3.1 Introduction

Once the informal security requirement has been documented, and the attacks and attributes identified, the next logical step in preparing a security problem definition is to perform a threat analysis to identify the threats represented by the potential attacks. ISO/IEC 15408 does not prescribe any particular methodology for identifying applicable threats. However, the methodology must identify all the threats perceived as relevant to the TOE in question.

Threat analysis and specification is usually more complicated and difficult than policy and assumption definition, and thus it is best to deal with it first. On the other hand, if the informal requirements have been mainly derived from policy decisions or mandatory requirements (see 9.2 above), it may be easier to draft the policy and assumption parts of the security problem definition first (see 9.4 and 9.5), then perform the threat analysis as described in this section, and finally revisit and complete the policies and assumptions. If policies and assumptions can readily be identified, they can then be used immediately to discount and exclude threats from further consideration, thus simplifying the threat analysis.

In order to perform a threat analysis, it is necessary to perform three activities:

a) decide on the analysis methodology to be used;

b) identify the participants required by that methodology;

c) apply the methodology.

These activities are discussed in turn in subsequent subsections of this clause.

9.3.2 Deciding on a threat analysis methodology

The best methodology to identify the applicable threats will depend on how the informal security requirement was derived. If the requirement was specified in terms of the results of a risk assessment, then a list of threats may already be available as one of the risk assessment outputs. Even if this is not the case, it may still be possible to identify the relevant threats from other existing and available information.

Unfortunately, in most cases sufficient information will not be available, and an additional threat analysis must be performed.

There are many possible methodologies that can be used to perform threat analysis. However, most developers of PPs and STs use one of three techniques:

a) threat tree analysis;

b) threat database search;

c) ad-hoc identification.

Of these, threat tree analysis is the best documented and established technique. It is based on the construction of decision trees, a well known problem decomposition technique widely used in risk management and reliability engineering (see, for example, [4] and [5]). The first description of its application to security threat analysis is recorded in [6].

Because it is a well established and well documented technique, threat tree analysis will not be described in detail within this Technical Report. However, in simple terms, it involves starting with a very general, abstract description of the complete set of threats potentially applicable to a type of IT product, and then introducing more detail in an iterative manner, refining the threat descriptions at each stage. The technique is referred to as a threat tree because the first abstract definition is considered as the root of a tree and each new level of subsequent refinement creates a set of new, more detailed, nodes connected to the root. Each of these nodes then becomes the root of a new sub-tree. Eventually, descriptions of leaf nodes will be sufficiently concrete to terminate the need for further refinement and be used as actual threats to be specified in the PP or ST. The tree also provides a rationale for the choice of threats included in the PP or ST, and gives confidence that no relevant threats have been omitted.

Recent proponents of the use of threat tree analysis include Bruce Schneier [7], and the Microsoft Corporation Trustworthy Computing Initiative. Indeed, a recent book from Microsoft provides a set of example threat trees that will match many types of software products ([8], Chapter 22) and which can be used as patterns to minimise analysis work for suitable TOEs. It is worth noting the caution from Microsoft that it can be difficult for non-security experts to build accurate and consistent threat trees (*ibid.*, box on page 128).

The second alternative, database search, is based on exhaustive examination of one or more predefined databases of generic threats, to see which entries match the identified attacks for the IT product in question. Suitable databases are available from many sources. Most national evaluation schemes will supply information concerning generic threats on request, and this is normally in the form of a searchable database.

Database search has a number of benefits and a number of disadvantages. The benefits are that a reasonably wide variety and range of threats will be considered, and that they are expressed and specified in a consistent way. One disadvantage is that there may be specialist threats to the particular product that are not covered, and therefore will not be identified. Also, threats descriptions in the database may be too general for applicability to the product in question to be readily identified. Finally, and most importantly, it may be found that too many threats appear applicable and a degree of arbitrary selection is subconsciously introduced.

The final alternative is to identify threats in an unstructured manner, based only on consideration of the IT product in question. This is best avoided - it is difficult for the developer or problem owner to “think outside the box”. Attackers may have more experience or more ingenuity in finding applicable threats.

If the security problem and its surrounding environment are both well defined, constructing a threat tree is usually the most effective approach. Where the problem is defined in general terms, or the environment is uncertain or arbitrary, a simple serial search of a threats database may suggest applicable threats more efficiently than methodological top-down analysis. This particularly applies to COTS product developers, who typically may not have much knowledge of the actual environments in which their products will be used.

If the informal security requirement was driven primarily by policies or mandated security features, do not be surprised if the threat analysis identifies no applicable threats that are not already countered by the required security attributes.

Depending on the threat analysis methodology used, and the origins of the informal security requirement, threats may be identified but subsequently discounted, or identified as duplicates of other requirements (such as policies). ISO/IEC 15408 does not require such threats to be documented *at all*, although it can then be very difficult to understand the SPD as a whole and in particular to modify it to reflect changes. This Technical Report strongly recommends that you *do* document discounted threats. The normal way to do so is as part of the assumptions section of the SPD (see 9.5).

9.3.3 Identifying participants

9.3.3.1 Introduction

Although previous versions of ISO/IEC 15408 only required that each threat was identified and explained, ISO/IEC 15408:2008 requires that each threat is described in terms of a threat agent, an asset and an adverse action – with the interpretation that “asset” is understood to include types of asset, since in the case of COTS products the actual assets to be protected are unknown to the person preparing the PP or ST.

Unfortunately, the results of risk or threat analysis and other forms of attack and attack path descriptions are rarely described in terms of agents, assets and adverse actions, and thus it is necessary to create the characterisation required by ISO/IEC 15408 from first principles, using the available threat and attack information.

9.3.3.2 Threat agents

ISO/IEC 15408 defines threat agents as “entities that can adversely act on assets”. There is no guidance on specifying threat agents, or the level of detail and precision required. When describing threats in PPs and STs, it is best to keep the threat agents used as simple as possible. One common approach, and the one recommended by this methodology, is to use a fixed list of five types of threat agent:

a) Attackers;

b) Authorised users;

c) Privileged users;

d) Administrators;

e) System owners and developers.

An *attacker* is a person who is not authorised to access assets protected by the IT product. This includes people who are authorised users, but have concealed their identity. Because they are unknown to the system owner, there is little deterrence unless their attack is detected and linked back to an identified person, for example, by telephone tracing or by visual identification by security guards.

An *authorised user* is a person who is authorised to use the IT product according to its security policy, and can access assets protected by the product with the permission of the owner of those assets. Authorised users are known to the system owner and are deterred from damaging assets by being held accountable for their actions.

A *privileged user* is a person who is authorised to use the IT product in a way contrary to its security policy, and can access assets without the explicit permission of the asset owner. Most system administrators must be privileged users. However, there are other types of privileged users – such as maintenance engineers, both hardware and software. Privileged users cannot be stopped by the IT product from causing damage, but can subsequently be held accountable for their actions.

By *administrator*, we mean people who are responsible for ensuring the correct operation of the IT product once installed in its operational environment. Administrators are therefore responsible for setting up controls to prevent damage to assets and also detecting when assets have been damaged. Administrators can be limited in what they do, but if they perform their actions incorrectly, assets may be damaged by others.

By *system owner and developer*, we mean those people who are responsible for the specification, design and implementation of a system or COTS product, but who do not necessarily use it to access the assets it protects. Although they cannot directly damage assets, if their decisions were incorrect, the product may be unable to adequately protect assets.

Using these definitions, a single individual may at different times fall into more than one of these characterisations – indeed, perhaps all. The distinction is through the type of threat they represent when acting as that type of threat agent.

The list above excludes one possible type of threat agent that may be relevant to some security problems – acts of nature (sometimes called “acts of God”), such as earthquakes, where there is no human agent involved. The usual approach is to treat such threats as being the responsibility of the system owner and developer, although they are not involved in formulating or executing any attack. In some cases, describing the related agent as “none” or “nature” may be clearer or more acceptable to the problem owner.

9.3.3.3 Types of asset

Assets are important to threat analysis and need to be properly identified. Most threat analysis methodologies can handle imprecision or overlap in players and adverse actions, but assets need to be distinct and well described. In consequence, this subsection offers a detailed methodology to identify the assets or types of asset that need to be protected by a particular IT product.

In the case of a system, it will often be possible to identify the precise assets to be protected, as this will form part of the definition of the system. In the case of a COTS product, the actual use of the product is often not known, and it is therefore only possible to identify the types of asset that the product is intended to protect.

Assets associated with IT systems usually fall into one of three classes:

a) information;

b) processes;

c) physical.

Information assets represent data that is of value to the owning organisation. Examples of possible types of information assets are:

- General data;
- System data;
- Specialist databases;
- Client data.

Specialist databases would represent information that is only of value to some users. Examples might be a personnel database (only of value to the human resources department) or a customer database (only of value to the order processing and marketing departments). Client Data might represent data not owned by the owner of the system and for which there is a special and relevant characteristic, a legal duty of care.

In the case of a system, it will normally be possible to identify the names and characteristics of the actual databases or other information assets to be protected.

In the simplest case, all data can be treated as being of equal value and at equal risk of attack, and represented by a single information asset, named something like “user data”.

However, it is often necessary to distinguish system data, i.e. data used by the TOE security functionality (TSF) of the TOE, from other data. If system data is modified or deleted, the TSF functions may operate incorrectly, and permit other types of attack, whereas if other data is modified, only the data directly involved is corrupted, the TSF continue to function, and will continue to protect other assets. It is quite common for these two information assets to be sufficient, one representing TSF data and the other all other data protected by the product.

Sometimes different types of TSF data may be susceptible to different attacks, or have different consequences if compromised, and thus required to be distinguished. Examples of distinct types of TSF data might be:

- TSF configuration data;
- The authentication data database;
- Audit records.

Sometimes very limited and specific forms of data that are susceptible to specialised attack may need to be distinguished; for example, cryptographic keys.

Process assets represent applications, where data is transformed or analysed. The distinction from information assets is that the associated data is of little value without the processing capabilities of the related applications. Examples of possible types of process assets are:

- Financial;
- Communication;
- Logistical;
- Manufacturing;
- Office Automation.

Financial applications might include payroll, investment management or accounts management. Communication systems might include e-mail or intranet/extranet information handling. Logistical systems might include order processing, warehouse control or resource scheduling. Manufacturing applications might include real-time process control. Office Automation might cover structured text processing.

In the case of a system, it will normally be possible to identify the names and characteristics of the actual processes to be protected.

In general, process assets are only susceptible to modification or denial of service attacks. For example, the functionality of the associated applications software could be altered, perhaps to remove authorisation checks or to alter financial processing. A single asset, called “applications software” or something similar, is usually sufficient to cover all processes.

Physical assets represent the actual information processing equipment used to support the information and process assets. Examples of possible types of physical assets are:

- Critical Network Infrastructure;
- Portable PCs;
- Data Centres.

It is very unusual for TOEs to offer protection of physical assets as part of the security problem – physical protection is either excluded, or provided by the operational environment and handled through assumptions. In consequence, it is therefore unusual for physical assets to appear in PPs or STs. However, there are applicable techniques, such as automatic closedown on power failure, that could offer protection to physical assets and in such cases physical assets might appear in the PP or ST.

It is important not to identify too many assets or types of assets. If two assets or types of assets have the same potential for attack and consequences of attack, they should be grouped together into a composite asset type. Many TOEs will protect only two types of asset, TSF data and user data. More than six types of asset is probably inappropriate for anything other than a TOE that is expected to offer very complex or individualised protection capabilities.

As part of the definition of the security problem, certain assets or types of assets may have been excluded from requiring protection. If this is the case, they should be listed separately: this information will be need later to explain why they have been excluded from the threat analysis.

9.3.3.4 Adverse actions

ISO/IEC 15408 provides no guidance on how adverse actions should be described. As for threat agents, the best advice that can be given is to keep the set of actions as simple as possible. One simple yet comprehensive set is:

- Improper access;
- Improper transmission of access rights;
- Denial of legitimate access;
- Non-accountability.

It has been found that this simple set covers pretty much all threats that are likely to be found in practice, although sometimes particular adverse actions may have distinct consequences which for clarity of explanation need to be described separately. There may also be other, specialised, types of adverse action that do not fall naturally into the groups above. This should be obvious from the informal security requirement and will again need to be treated separately.

An alternative approach is to describe adverse actions in terms of the consequences of a successful attack, e.g. loss of confidentiality. This approach was often used in the past. However, it can be unnecessarily specific and limiting in scope. It is no longer often used.

9.3.4 Applying the chosen threat analysis methodology

Once a threat methodology has been selected, and the necessary information to apply that methodology has been prepared, the next step is to apply it to generate a list of applicable threats.

In practice, many possible threats can quickly be discounted. There are two particular techniques that are very useful – identifying excluded or tolerated threats, and identifying threats already covered by policy.

Many types of threat will have already been discounted as part of the definition of the informal security requirement, either because they have been excluded from the scope of the IT product, or a decision has already been made to tolerate them because the impact of associated risks is low, or they have been transferred to a third party (e.g. an insurer).

Exclusion is common in the context of COTS products – for example, the vendor of an operating system may decide not to include anti-virus (AV) protection within the product, assuming that the purchaser will wish to buy a supplemental specialist AV product, or will use the product in an environment that is isolated from infection.

Tolerating threats is usually found in the systems context; it requires assets to be valued, something a COTS product manufacturer cannot do.

The relevant information to discount threats is usually obvious from the list of things that the product need not do. If not, it needs to be confirmed and then added to that list. It should also be recorded in the form of an assumption (see 9.5).

In many IT products, a decision will have already been made to include security functionality, independent of the analysis of actual threats. It is common in the case of COTS products – for example, an operating system vendor will normally include user identification and authentication functions, even if the product is designed for single user situations.

If this mandated functionality will counter a particular type of threat, that threat need not be investigated further to see if it is actually applicable; protection will be provided regardless.

The relevant information to ignore threats is usually obvious from the list of attributes that the IT product must possess. If not, it needs to be confirmed and then added to that list. It should also be recorded in the form of a policy statement (see 9.4).

All remaining threats must be identified and considered, and a full list of applicable threats produced, describing each threat in terms of agents, assets and adverse action.

Some threats may be applicable to a particular system, but it has already been decided as part of scoping the security problem that they will be countered by security controls within the operational environment. It may only be possible to counter some threats by measures in the environment (for example, where physical protection is necessary). These threats still need to be listed, but it is worth making a note with the entry that they will generate environmental objectives; this information will be very useful later.

However do not prejudge how threats will be countered if it could be done by either the TOE or its environment. This would take away the ability to make design trades later when controls are being selected and designed.

Using older versions of ISO/IEC 15408, threats to the development of the IT product (i.e. its development environment) were included within the threat analysis. However, where ISO/IEC 15408:2008 is used, this is no longer required. Do not include such unnecessary information within your threat analysis. It will only confuse the evaluators.

9.3.5 Practical advice

Threats indicate possible ways that the IT product might be attacked. Therefore they should be worded as such. The best way to do this is to use a verb form such as “may”. For example:

T.UNAUTH An unauthorised person may attempt to access and use TOE resources.

It helps to start each threat description with a name for reference purposes. By convention, most PP and ST authors start threat names with “T.” to assist identification. Descriptions should be kept short and to the point.

Methodologies, whether the one described in this clause or one of your own choice, must not be used blindly. They must be adapted and interpreted to meet the requirements of a particular security problem. Do not be afraid to go back and start again using a different approach if a particular form of categorisation is not working out in practice.

Threats can be combined if their agents, assets and adverse actions are similar. This will reduce the size of the threat list and save time later, since the same controls will often be used to counter them. Equally, where a threat has markedly different impacts depending on factors like threat agent or asset involved, it will be clearer and save time later if the threat is split into multiple threats that are more specifically worded.

Information indicating that threats can be discounted is often expressed indirectly. For example, consider the statement:

Administrators can be assumed to be non-malicious, trustworthy and competent.

This is expressed in terms of a threat agent, and effectively discounts most types of threats normally associated with that type of agent. Some of these types of threats are specific to administrators and can therefore be fully discounted. Other types of threats will still apply, but can be restricted to other applicable threat agents only, e.g. ordinary users. Do not forget to add the assumption that reduced the scope of these threats to the list of assumptions.

In some cases it may not be possible to identify threat agents or adverse actions – only that the associated risk is unacceptable. An example would be failure of an underlying abstract machine to implement its associated security model. In these cases, it is pointless to create characterisations based on guesswork or imagination. The threat is unacceptable by definition of the security problem, and should be identified and justified as such.

Once a final list of threats has then been prepared, it should always be checked for completeness and consistency. If a threat has been broken down by type of asset or type of threat agent, are all possibilities covered? Are similar threats treated in a similar manner? If not, is there a good reason? Although inconsistencies and omissions may well be detected later in the preparation of the PP or ST, checks at this stage will save time and reworking later.

It is possible that threat analysis may identify no threats to be listed as applicable to the TOE. This can happen, for example, in PPs that are designed to meet general corporate or government policies and nothing else. This is perfectly acceptable in ISO/IEC 15408 evaluation; the threats section should be left blank, with an indication that no specific threats were identified.

Historically, successful general-purpose PPs have specified few or no applicable threats. If you are producing a PP intended for use in multiple contexts, and you have identified a large number of applicable threats, question whether you are unconsciously assuming a context that is unrealistic or unnecessarily limiting.

9.4 How to identify and specify policies

The security problem definition must also contain a list of applicable *organisational security policies* (OSPs) with which the TOE must comply. Compared to threats, policies are generally much easier to identify and describe. If you are using our recommended methodology, you will already have a list of security attributes or features that the IT product must possess. Each of these can be reworded to become an OSP.

Policies are statements of things that the IT product must do, regardless of consideration of threats or other matters. Therefore they should be worded as such. Standards written in English use the verb form “shall” to indicate requirements of this type. Most English speakers find this unnatural, and the verb form “will” is perhaps to be preferred. Thus an example of a clear and well worded policy might be:

P.IDAUTH Administrators will authenticate themselves before accessing any TOE functions or data.

As for threats, it helps to start each policy with a name for reference purposes. Descriptions of policies should be kept short and to the point. By convention, most PP and ST authors start policy names with “P.” to assist identification.

In ISO/IEC 15408, policies are normally referred to as Organisational Security Policies, or OSPs for short. This can be confusing – some OSPs may only apply to one system to be covered by a PP or ST, rather than all systems within the owning organisation. This Technical Report usually uses the simpler term “policies”.

Most applicable policies should have been identified during identification of the informal security requirement, or during threat analysis. However, a final check should be made to identify any other policies that are relevant to the security problem.

Policies are used to specify:

- Mandatory security functions to be incorporated within the TOE;
- Mandatory technologies/techniques to be used to implement particular security functions (which implicitly requires those functions to be present).

Policies can also be used to replace threats. This is appropriate if:

- It is not certain that a particular threat exists, but a policy decision has been made to protect against it regardless;
- A policy decision has been made as to how a particular threat will be countered, e.g. by specifying:
    - what controls will prevent a successful attack; or
    - what will be done if an attack occurs;
- A policy decision has been made to adopt a particular approach to countering a number of related threats.

However, there is no value in replacing a threat with a policy unless there is some additional information represented in the policy that is not implicit in the statement of the threat.

Policies identified during this final check may require changes or reworking of previous security problem definition activities, e.g. to delete threats that are now covered by policies.

In practice, most policies are easy to identify and express clearly. However, there are some common problems that should be noted.

Policy statements are sometimes misused to express requirements for things that the TOE must not or cannot do, but which instead must be enforced by the operational environment of the TOE. If a requirement cannot be implemented by the TOE, the correct way to specify it is as an assumption concerning the operational environment (see 9.5). If a proposed policy cannot be enforced by the TOE, the operational environment, or by the two working together, then the policy statement is either meaningless or unachievable.

During the course of specifying the security problem and its solution, the boundary of the proposed TOE may need to change, to transfer functions from the TOE to its operational environment or vice versa. This may cause policies to become assumptions or assumptions to become policies, or it may require policies or assumptions to be re-specified to take account of the new TOE boundary. Similarly, in composed TOEs that are broken down into several components addressing different security problems, an assumption for one component is often implemented by another as a policy requirement. In such cases, careful wording of the policy statements will enable them to be reused in the other SPDs as assumptions, ensuring compatibility and easy consistency checking.

Sometimes it is not clear during preparation of the security problem definition whether a policy will be implemented by the TOE or by its operational environment. This is acceptable; it can be resolved during definition of the security objectives when the requirements for security functionality are clearer. Both TOE objectives and environmental objectives can link back to policies. A policy may even be partially implemented by the TOE and partially by the environment.

Not all security problems require policies. This is perfectly acceptable in ISO/IEC 15408 evaluation; the policies section should be left blank, with an indication that no applicable policies were identified.

9.5 How to identify and specify assumptions

Finally, the security problem definition must contain a list of applicable *assumptions* that limit or exclude the security features required within the TOE. If you are using our recommended methodology, you will already have a list of security attributes or features that the IT product need not possess. Each of these can be reworded to become an assumption about the environment or intended usage of the TOE.

Assumptions are statements of things that the IT product need not do, regardless of consideration of threats or other matters. They should therefore be worded as statements of fact. An example of a clear and well worded assumption might be:

A.PHYSICAL The TOE will be located in a physically secure location.

Assumptions have two uses:

- To indicate that a particular control or type of control will be provided by the operational environment, and not the TOE;

- To indicate that particular threats or type of threats can be discounted, because in the content of the assumed operational environment, they will not exist or are not important.

The first of the above types is best expressed using the verb “will”, as it implies a control must be provided, even if not by the TOE. The second form is best expressed using an active, present tense verb such as “is”.

Keep assumptions about controls provided by the environment distinct from assumptions about discounted threats, as the former is required by ISO/IEC 15408 and the latter an addition recommended by this Technical Report to simplify showing security objectives cover all applicable threats. This will be explained later (see 10.2).

Every assumption should be given a short name for reference purposes. Descriptions of assumptions should be kept short and to the point. By convention, names of assumptions start with “A.” to assist identification.

In practice, it is more difficult to express assumptions clearly and positively than it is policies or threats. Avoid the temptation to use verbs such as “may” or “should”: assumptions are statements of fact.

Assumptions about the operational environment need to be separated into the three areas of:

- physical protection;

- personnel and procedures;
- technical functionality outside the TOE.

ISO/IEC 15408 refers to “physical, personnel and connectivity aspects of the environment” (ISO/IEC 15408-1, subclause A.6.4). However, practical experience has shown that this is not sufficient. Many assumptions about external technical controls do fall naturally under the heading of connectivity, for example:

A.INTERNET The TOE will be isolated from the Internet.

However, other assumptions about technical controls are often necessary. For example:

A.NO_DEV_TOOLS No tools will be present in the operational environment of the TOE that permit ordinary users to add new functionality to the system.

In many cases policies and threats will be partially handled by the TOE and partially by the environment. For example, technical controls within the TOE may need supporting procedural or physical measures to be present in order to work effectively. The need for such supporting measures in the environment must be identified and expressed as assumptions.

Assumptions are not tested during evaluation; they are treated as always valid and true. However, they are helpful in showing consistency and completeness. Where threats have been identified by a methodological approach, assumptions may be needed to show complete coverage in the rationale. A threat may be partially discounted and partially countered. In this case the assumption is needed when tracing back the security objectives for the countered part back to the threat to show that complete coverage is provided.

Many assumptions will have been identified during specification of the informal security requirement, or during threat analysis. However, a full investigation should be made as part of this stage of security problem definition to identify any other relevant assumptions. When a decision is made that a policy will be implemented, or a threat countered, by the environment, this must always be recorded as an assumption. These assumptions should be worded to reflect the policies and threats in question, as they will generate objectives for the environment that will need to match those policies and threats.

One assumption can often be used to counter multiple threats that are related in some way. If a threat tree approach has been used, where multiple detailed threats all to be countered by the environment share a common hierarchical node further up the tree, the assumption can be expressed at the level of the shared node. For example, if all threats resulting from adverse actions by administrators are discounted, this can be expressed in one single assumption:

A.NO_POOR_ADMINISTRATION Administrators have the necessary skills, training, time and resources to perform all their allocated administrative functions, and perform all those functions correctly.

When formulating assumptions, a good test for a well formed and necessary assumption is that if the statement is untrue, the TOE could be successfully attacked.

Separating assumptions by type will be helpful when identifying and specifying security objectives. Assumptions about personnel, procedural and physical security should be separated out first. The next category should cover assumptions related to security functionality provided by the IT operational environment. Finally, the assumptions about discounted threats. These should be kept fully separate as these do not generate objectives at all.

There may be some security problems that do not need any assumptions. This is perfectly acceptable in ISO/IEC 15408 evaluation; the assumptions section should be left blank, with an indication that no required assumptions were identified.

9.6 Finalising the security problem definition

The last stage of SPD production is finalising the SPD specification. This involves two tasks:

- Preparing a complete list of all threats, policies and assumptions.

- Performing consistency and completeness checks to confirm the SPD specification accurately represents the security problem or problems addressed by the informal security requirement.

There is no requirement in ISO/IEC 15408 to provide an SPD rationale; the statement of threats, policies and assumptions expressed in the SPD is treated as axiomatically correct for the purposes of evaluation. However, it is strongly recommended that a rationale is produced, linking each element of the SPD back to the informal security requirements, and showing that coverage is complete, without duplication and without redundancies. If requirements change, or complications are found later on, the rationale will make SPD reworking much simpler and reduce the risk of introducing errors.

Similarly, there is no requirement in ISO/IEC 15408 to identify threats that have been discounted or ignored. Once again, this information is extremely useful if circumstances change and the SPD has to be reworked. This Technical Report recommends that appropriate assumptions about such threats are always included. However, put them in a clearly marked separate section of the SPD, distinct from any assumptions about the operational environment. This will signal to evaluators validating the SPD that they must be ignored when tracing back security objectives.

Consistency and completeness checking involves checking that all constraints and requirements found whilst scoping the security problem have been reflected in policies or assumptions, and that all identified threats have been countered or discounted in some way. Similarly, all policies, threats and assumptions listed in the SPD should be traced back to aspects of the original informal security requirement. Creating cross-reference tables is often an efficient and easy way to show that consistency and completeness exist.

Assumptions and policies may sometimes appear to conflict, i.e. a firm policy requirement “will do X” may appear to be contradicted by an assumption “need not do X”. On inspection, it will usually be found that there is no actual conflict, the TOE is expected to address part of some identified security concern but not the whole. Greater explanation and precision of wording in describing the actual requirement is needed, and will resolve the apparent inconsistency. If there is a real conflict, it must be resolved by re-examining the informal security requirement to establish what was actually wanted.

10 Specifying the security objectives

10.1 Introduction

This clause provides guidance on the identification and specification of security objectives in an ST or PP, the requirements for which are described in ISO/IEC 15408-1, clauses A.7 and B.7 respectively. As for security problem definition, B.7 consists merely of a pointer to A.7, strongly implying that the expected contents are identical in both cases. As for the security problem definition, the validation requirements in ISO/IEC 15408-3 are the same in both cases.

The *security objectives* provide a *concise statement of the intended response to the security problem* (ISO/IEC 15408-3, subclauses 9.4.1 and 10.4.1). This should not be misinterpreted – the response is really the specification of the security functional requirements (see clause 11). The security objectives work best if they are expressed as an overview and structure of the security functionality required, providing a link between the detail of the SFRs and the abstract problem definition of the SPD. In other words, having stated in the security problem definition what the security issues are, you now need to give an indication of how they will be addressed by the TOE and its environment.

ISO/IEC 15408 requires two different types of security objectives to be specified:

a) Security objectives for the TOE, which will be satisfied by technical (IT) countermeasures implemented by the TOE;

b) Security objectives for the environment, which are to be satisfied by either technical measures implemented by the IT environment, or by non-IT (e.g. procedural) measures.

This is illustrated in Figure 2 below.

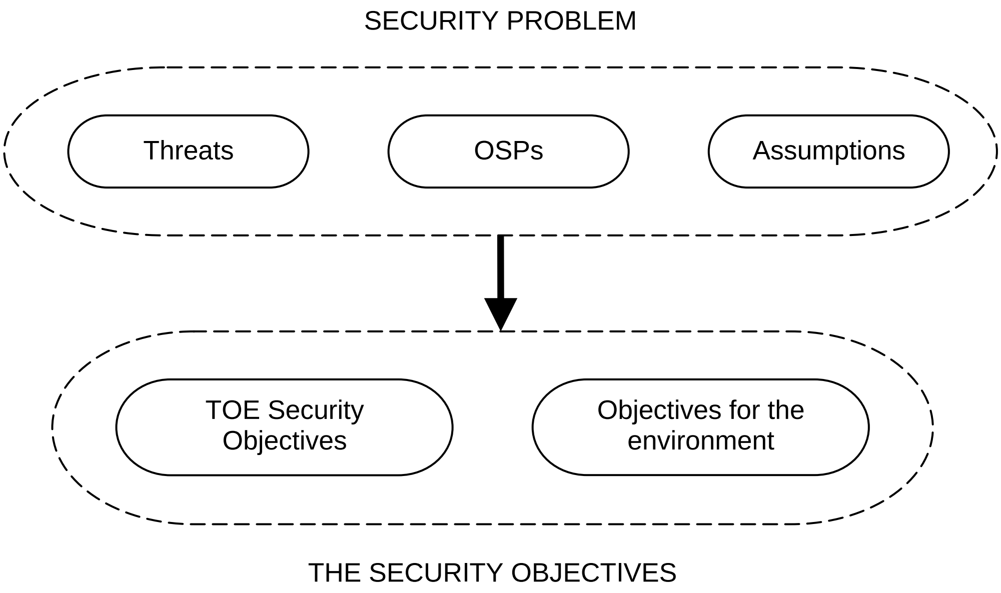

Figure 2 — Role of the security objectives

All PPs and STs have to specify security objectives for the operational environment. Low assurance PPs and STs (see clause 15) do not have to specify security objectives for the TOE, and the security objectives for the operational environment are treated as axiomatic, i.e. they do not have to be linked back to a security problem definition.

The remainder of this clause assumes both types of objectives are required, and are linked back to a security problem definition.

Security objectives should be worded as requirements. They should consist of short, clear statements that together define a high-level solution to the security problem identified in the related SPD. In English, the verb form “must” is a good way to word objectives.

ISO/IEC 15408 does not assume or mandate any particular process or methodology for preparing the security objectives; you can use any method you like. Of course, if you are new to the process of developing PPs and STs this is not helpful. This clause therefore includes a detailed description of a simple methodology that has been tried and tested in practice and found to work in a variety of organisations and environments. It is based upon a series of steps, performed in sequence:

a) Structuring the list of all threats, policies and assumptions to be covered by the objectives;

b) Identifying the objectives for the non-IT operational environment;

c) Identifying the objectives for the IT operational environment;

d) Identifying the objectives for the TOE;

e) Producing an objectives rationale linking back to the identified threats, policies and assumptions.

Each of these steps is described in following sections. They are usually best performed in the order given above.

Of course, this is only one possible approach to identifying your objectives. Other equally valid approaches exist. In your particular circumstances, this may not be the simplest, fastest or easiest approach. Do not be afraid to experiment.

Because of the pivotal role played by the security objectives in the PP or ST, the question of what level of detail is appropriate in a statement of security objectives is important. ISO/IEC 15408 gives a strong hint by saying (as pointed out above) that security objectives are intended to be *concise.* In practice, you need to strike a balance between the following two considerations:

a) The security objectives should help the reader to understand how the security issues identified in the security problem definition are to be addressed by the TOE, without delving into implementation detail except where this has been mandated in the SPD; ideally, the security objectives for the TOE should be implementation-independent. The focus is thus on *what* the solution intends to achieve rather than *how* it is achieved.

b) At the same time, you should ensure that the security objectives as defined do not just repeat the information contained within the threats and OSPs of the security problem definition, albeit in a slightly different form.

One test of whether you have pitched your security objectives at the correct level of detail will come when you construct the rationales for the security objectives and the security requirements. If one rationale is trivial whilst the other is large, complex and difficult to understand, it is likely that your security objectives are either too detailed or too abstract, depending on which step is too complex.

As will become clear in the next clause of this Technical Report, a well-defined set of security objectives for the TOE will help ensure that the security functional requirements selected to meet them are not excessive. This in turn will serve to minimise the cost and timescales of the TOE evaluation.

10.2 Structuring the threats, policies and assumptions

The first task is to structure a complete list of all applicable threats, policies and assumptions from the SPD.

Remember that some threats may be relevant to the TOE, but risk analysis or consideration of the environment may have decided that they are can be discounted or ignored. If you have followed the methodology recommended by this Technical Report, you will have included these threats within the SPD, but also recorded assumptions that identify them as not applicable. Such threats do not generate security objectives, so your first step is to identify them and their related assumptions, and exclude these threats and assumptions from further consideration. Make sure that it is obvious from your SPD that these threats have been excluded in this way.

The remaining threats, policies and assumptions should then be separated by type:

- Those relating to the non-IT operational environment;

- Those relating to the IT operational environment;

- Those related to TOE functionality.

This is usually easier than it might appear: a policy requiring physical controls can only apply to the non-IT environment; a threat representing a possible attack directly on the TOE belongs to TOE functionality. Note that assumptions can only apply to the operational environment sections. Where a policy or requirement appears to span several areas, it should be subdivided and one part assigned to each.

For example, a threat T.EAVESDROP might be split into two:

- T.EAVESDROP (communications), assigned to the IT operational environment;

- T.EAVESDROP (internal), assigned to TOE functionality.

If in doubt, split the policy or threat concerned into multiple areas. Unnecessary entries are easy to delete later. On the other hand, missing entries may cause objectives to be missed, and are much harder to detect during PP/ST validation.

10.3 Identifying the non-IT operational environment objectives

Generally speaking, it is easier to define the objectives for the operational environment than for the TOE, and the non-IT objectives are easier to define than the IT environmental objectives. So it makes sense to work on the non-IT operational environment objectives first.

The first step in identifying these objectives is to take all the assumptions assigned to the non-IT operational environment and reword them on a one-to-one basis into corresponding objectives (there is guidance on how to do this later in this section). Environmental objectives are not analysed further within the PP and ST, or during evaluation, so there is little point in identifying commonality, generalisation, overlap etc., provided that the stated objectives are clear, and clearly defined.

Then devise and add any further objectives necessary to meet aspects of threats and policies that have been assigned to the non-IT operational environment, once again rewording the threats or policies concerned as objectives, but without expansion or explanation. Identifying suitable wording is again usually straightforward. If not, the categorisation techniques used for the more difficult area of TOE objectives and described in section 10.5 below can be used.

Other security objectives for the non-IT operational environment may include:

a) establishment and implementation of procedures to ensure that the TOE will be used in a secure manner (and in particular in accordance with the environmental assumptions);

b) objectives for education and training of administrators and users in sound security practices.

These may be more difficult to identify at this stage, as they support security objectives for the TOE. If they are obvious, include them now. If not, later stages of the methodology will revisit the environmental objectives and add them in.

Environmental objectives are often given identifying names starting with “OE.” This helps to make a clear distinction from TOE objectives, which conventionally start with “O.” They should be clearly worded to indicate that the measures implementing the objective will be procedural or physical; if necessary state “the non-IT environment” explicitly in the description of the objective.

Environment objectives derived from assumptions are best worded unchanged from the assumption wording, i.e. as factual statements. For example:

OE.RESIDUAL Magnetic media are degaussed or shredded prior to final disposal.

Objectives derived from threats and policies should be worded as requirements. For example:

OE.AUD_REVIEW Operations staff will review audit trails for exceptions and unusual patterns of activity at regular intervals.

Most non-IT operational environment objectives will be derived from assumptions. Objectives that are derived from consideration of threats alone may indicate assumptions are missing from the security problem definition. Check the SPD, and revise if necessary.

For convenience, single objectives can be defined that cover several related assumptions, or an assumption and related threats, or policies and related threats. It is worth combining such elements together if the overall result is clearer. If not, do not bother.

Satisfying the non-IT operational environment objectives will be the responsibility of the organisation that uses the IT product in question. It is very important to check at this stage with the people responsible for system operation (or the marketing department in the case of COTS products) to ensure that these objectives are realistic and achievable. If not, it is better to know of the problems now rather than later, while the objectives can still be altered or the threats and policies handled in different ways.

10.4 Identifying the IT operational environment objectives

The techniques used to identify and specify the objectives for the IT operational environment are identical to those for the non-IT objectives described in 10.3 above. However, it is important to keep them separate from the non-IT objectives, because IT environment objectives could become TOE objectives if the TOE boundary changes later during TOE specification and design.

By convention, objectives for the IT operational environment are also identified by giving them names that start with “OE.”. Similarly, they should include “the IT environment” within the description, or otherwise make it clear that they will be implemented by technical means outside the TOE.

In earlier versions of ISO/IEC 15408 it was permitted to specify security requirements for IT environment objectives in order to define and explain exactly how they were supposed to be achieved. This is not permitted in ISO/IEC 15408:2008. However, there are other techniques, such as the use of application notes, that can be used to record constraints on the implementation of objectives.

In a composite product, objectives for the IT environment of one domain will become objectives for the TOEs of other domains. Such objectives should be very carefully worded, to ensure that the correspondence can easily be identified.

10.5 Identifying the TOE objectives

The TOE security objectives are the most important and the most difficult objectives to express well. Unlike environmental objectives, they are used as the justification for security functional requirements. It is therefore important that they are well worded, clear in their intention and provide good traceability between detailed security requirements and the security problem; it is not sufficient just to reword the security problem or list specific security requirements.

The methodology proposed in this clause organises TOE objectives on the basis of broad areas of security functionality, chosen to link well with the organisation of functional components into families and classes within ISO/IEC 15408-2. Breadth and depth is dealt with through the concept of main and subordinate objectives within each area. Each main objective sets a broad strategy for that aspect of security, a “best practice” target; the subordinate objectives deal with the specific points of detail that appear in any security problem but which if not treated properly can easily obscure the “bigger picture”.

Using this methodology, the first step in defining these TOE objectives should be to reorder the list of applicable threats and policies assigned to TOE functionality at the start of this step in order to place related threats and policies together. There should be no assumptions relating to TOE functionality, as assumptions are only made about the operational environment. If any assumptions have been assigned to this heading, your response is simple – investigate and fix.

The best form of grouping for a particular PP or ST will depend on the nature of the related TOE. However, it will always be helpful later when generating the SFRs if the grouping is related to the internal structure of ISO/IEC 15408-2.

The methodology used by this clause proposes a simple set of seven headings, under which all threats and policies are grouped. This methodology has been tried and tested in practice, and found to work for many types of TOEs. The headings are:

a) Access control (objects, attributes, operations, rules for access).

b) User management (user types, identification, authentication).

c) TOE self protection (detection of malfunction, trusted recovery etc.).

d) Secure communication (establishing communication links, link properties, rules).

e) Audit (event logging, reaction, incident management, analysis).

f) Architectural requirements (required properties and constraints).

g) Other functions (anything not falling easily under these headings – e.g. trusted time source, random number generation).

There is a deliberately close relationship between these suggested headings and the structure recommended in clause 12 of this Technical Report to identify and specify security functional requirements. Although the security objectives may have any structure and method of organisation that you like, in general the breakdown recommended above will simplify cross-checking and generating arguments concerning completeness and consistency later. Of course, there will always be particular TOEs when a different type of organisation will be clearer and easier to work with later. The important thing is to think about structure at this stage and to pick an appropriate approach.

The next step is to write down a simple definition of the type of security service or security protection required in each of your selected areas needed to meet the overall needs of the security problem. Rather than trying to analyse and generalise the security problem definition, it is better to return to the informal security requirement from which the SPD was derived. It is usually obvious from the informal security requirement what the major security functions in each of these broad areas should be. Some areas may not be mentioned, or may be explicitly identified as not relevant; ignore these at this stage.

This list of services should then be compared against the grouped list of threats and policies. For each service, decide which threats and policies are relevant. At the end, put any threats and policies remaining under the “other” service.

Next, divide the threats and policies associated with each service into general and specific requirements. General requirements should apply to all aspects of the service definition, specific requirements to special cases.

Finally, reword the service definition into a positive statement that addresses the general requirement. This becomes the main objective for that service. Reword each specific requirement into a related but separate subordinate objective for that service.

Threats can be countered by an objective that stops the threat by removing or blocking one of its necessary components. Examples of this are removing the ability of the threat agent to execute the adverse action, moving, changing or protecting the asset so that the adverse action is no longer applicable, or eliminating the threat agent (e.g. by introducing an environmental objective for physical access controls). Threats can also be handled indirectly. Examples of this are enforcing accountability through auditing actions, better training to stop accidental user errors, taking frequent backups so that lost or damaged assets can be easily restored.

Not all threats can be protected against. Sometimes the best course of action is to detect a related incident, and generate an alarm or audit log entry. This type of design decision will have to be made at this time. When detection is chosen as the response, this will generate the need for an audit service to respond to incidents.

During the specification process it may be necessary to reassign threats and policies. As services become better defined, particular threats or policies may fit more readily under a subordinate objective rather than the main objective or vice versa, or they may even fit better as part of another service. The process often identifies related objectives for the operational environment that have previously been missed; for example, there will be a need for administrators to respond to alarms, if alarms are chosen as the response to a particular threat. In some cases, design decisions may even move protection for particular threats or policies from the TOE objectives to the operational environment completely, or vice versa. These changes are to be expected; it will be necessary to iterate several times until a clear list of objectives is obtained covering all areas.

As well as expressing general protection requirements (linking directly to a main objective), policies in particular are sometimes used to constrain the nature of the associated technical solution. This type of constraint should be expressed as a subordinate objective, linked to the general requirement.

Some threats will link directly to a specific subordinate objective that counters that threat and no other. In this case, word the objective to directly reflect its source. This will ease later traceability – both in the rationale linking objectives back to the SPD, and for the understanding of readers.

A subordinate objective may address several threats and policies. For example, many PPs and STs have an object reuse objective as a subordinate objective in the area of resource management. This is worth separating out from other aspects of resource management as there is generally little overlap in terms of the threats addressed. However, there is no need to divide the objective further by the different types of resources that might need to be cleared – although different types of resources might be handled in different ways e.g. some threats to RAM do not apply to magnetic media. The distinction will become clear at the security requirements specification stage, when different SFRs will be selected as mechanisms for different resources.

A further useful distinction in defining subordinate objectives is by the type of control required. Controls can be preventative (stop an incident taking place), detective (recognise an incident has taken place) or corrective (fix the consequences of an incident). It is worth having different subordinate objectives for each type if the treatment of threats or policies needs actions of more than one of these types in response. This is often the case if the description of the security problem requires defence in depth, or if the main objective for a service will only reduce or mitigate a threat rather than blocking it.

An example of a *preventive* security objective is the following, which identifies the need for identification and authentication of users of the TOE:

The TOE will ensure that each user is uniquely identified, and that the claimed identity is authenticated, before the user is granted access to the TOE facilities.

Access control and information flow control security objectives also fall into the *preventive* category. Where the security concerns indicate that the TOE should enforce more than one access control or information flow control policy, it is recommended that you identify distinct security objectives for each policy. Such an approach will help simplify the security requirements rationale.

An example of a *detective* security objective is the following, which identifies the need for the TOE to provide a non-repudiation of origin capability:

The TOE will provide the means by which a recipient of information can generate evidence which can be used as proof of the origin of that information.

An example of a *corrective* security objective is the following, which identifies the need for the TOE to respond to detected intrusions:

The TOE will, upon detection of events that are indicative of an imminent security violation, take appropriate steps to curtail the attack with a minimum of disruption to the service provided to other TOE users.

At this point it will be necessary to revisit the statement of security objectives for the operational environment to see if there any security objectives relating to management activities that need to be added to ensure that the security services to be provided by the TOE are effective. In some cases, the required management activity is obvious, and can be immediately expressed in the form of a (non-IT) security objective. In other cases the required management activity may depend on the detailed security requirements used to implement the TOE security objectives. For example, an "identification and authentication" user binding security objective might be implemented by user passwords. This would imply a requirement for users to ensure their passwords are not disclosed to other individuals, which would properly be expressed as a security requirement for the non-IT environment. Do not be upset, or surprised, if you miss this type of implicit requirement at this stage. It will become obvious when you define the SFRs, and the statement of security objectives can be updated at that time.

Where possible, the security objectives should aim to informally quantify the minimal effectiveness expected, thus leaving little doubt as to what level of effectiveness must be justified in the PP or ST rationale. Quantities may be stated:

a) in relative terms, e.g. to environmental conditions or to a previous situation;

b) in absolute numeric terms.

Clearly, specifying absolute numeric values is the most precise option, but is also the most difficult to assess in terms of effectiveness.

Do not expect one to one correspondence between objectives and threats or policies. Often a main objective required to handle a policy will also counter many of the threats related to that service. Also, threats and policies may have to handled differently for different types of asset, and need different subordinate objectives for each asset type.

There are other techniques that can be used to identify security objectives. A simple approach, which can work well for small SPDs, is to simply to generate one objective per threat or policy, reflecting its wording, and with substitutions for specific assets, threat agents etc., if these are not clear from the wording of the related threat or policy in the SPD.

TOE objectives are generally given identifying names starting with "O." rather than "OE." to distinguish them from environmental objectives. They should be clearly worded to indicate that the measures implementing the objective will be part of and enforced by the TOE.

TOE objectives are sometimes worded to start "the TSF must" or "the system must". The TSF is that part of the TOE that implements the SFRs. This distinction is made for practical reasons, to reduce the amount of the TOE that has to be examined during evaluation. The use of the term "TSF" is therefore strictly correct; for any objective, that part of the TOE that implements it must be part of the TSF. However, this is somewhat a circular argument and also a little confusing as these objectives are usually referred to as "TOE security objectives", not "TSF security objectives". Saying "the system" is also confusing. It could be interpreted to include objectives implemented by the operational environment. If this is intended, it is much better to say "the TOE or its environment". Note that design decisions must separate such objectives into objectives for the TOE and for its environment before the objectives are finalised.

10.6 Producing the objectives rationale

The final step in defining the security objectives is to produce a rationale, tracing the objectives back to the threats, policies and assumptions in the SPD to show that they are all necessary, and also showing that all aspects of all threats, policies and assumptions in the SPD are covered by the objectives, or have been excluded from further consideration. For all but low assurance evaluations, this rationale is required by ISO/IEC 15408, and checked in PP/ST validation.

A simple way to produce the rationale is to prepare tables of the relationships between the SPD elements and the objectives and vice versa, and check for any inconsistencies, gaps or overlaps. Where threats, policies or assumptions are handled by multiple objectives, there is usually a simple discriminant that can be attached to the SPD element to show which parts are countered by which objective – see the example in 10.2 above. Including this in the table will make the mapping much clearer and easier to understand.

Assuming that each security objective can be linked back to at least one threat, policy or assumption, the table should show immediately that each security objective is *necessary*. Of course, this does not guarantee that there are no redundant security objectives, since other security objectives may also link back to the same threats, policies and assumptions, and already provide adequate coverage. However, this can be determined as part of establishing the second validation requirement, *sufficiency*.

Sufficiency has to be shown by providing informal arguments to supplement the cross-reference information. For each non-discounted threat, you need to argue why the related security objectives, taken together, will provide an effective countermeasure to the threat as defined. Note that attacks based on these threats do not necessarily need to be eliminated completely; it may be sufficient to detect or recover from successful attacks, or to reduce the likelihood of attack to an acceptable level. All that is required is an effective countermeasure within the context of the SPD.

Similarly, for each identified OSP or environmental assumption, you need to justify by providing informal arguments that the related security objectives are sufficient either to provide complete coverage of the OSP, or to uphold the assumption.

Remember that assumptions included within the SPD to identify threats that can be discounted or ignored do not generate security objectives and should therefore not appear in the objectives rationale.

If a PP or ST claims compliance with other PPs, the rationale will need to show that the security objectives for the TOE are consistent with the statements of security objectives within the referenced PPs. If those security objectives are worded in a similar way to your own, you may be able to show through straightforward mapping that your objectives match and cover all their objectives. Indeed, if the PP is one that requires strict conformance, the wording must be identical and the evaluators will therefore ignore whatever you put.

However, it is possible that the referenced PP objectives may be structured or worded very differently, such that there is no simple correspondence. In this case, you will need to show that the security objectives for your TOE also satisfy the requirements of the security problem definition sections within the referenced PPs. From this, you can argue that your objectives provide the same coverage as their objectives and are therefore consistent.

It may be impossible to generate convincing sufficiency arguments where a PP or ST claims compliance with other PPs, and the security problem definitions in the referenced PPs do not explicitly cover all the threats in your SPD. There is no solution to this – COTS products conforming to the referenced PP may be perfectly suitable for your purposes; however, their claimed PP compliance will not prove it. You may be able to establish that such products do meet your requirements by looking at the threat sections of their STs and establishing that they do consider and cover all your relevant threats.

11 Specifying extended component definitions

When attempting to specify the security functional requirements and the assurance requirements, the author of a PP or ST may not be able to correctly specify the requirement even when using the freedom he has to refine existing components from Part 2 or Part 3 of ISO/IEC 15408. In those cases the standard allows the definition of extended components. This section is intended to provide some guidance for the specification of extended components.

Before providing this guidance, one general advice: the definition of extended components should be avoided whenever possible! Using extended components makes it harder to compare different products based on the security functional and assurance requirements the products satisfy. Instead, one should first attempt to use existing components from ISO/IEC 15408, potentially with refinements. Only in cases where this is not possible, extended components should be used.

Refinements can quite often solve the problem when a component from ISO/IEC 15408 does not seem to be able to address a specific requirement one wants to express in a PP or ST. For example, when the requirements for user authentication differ for different types of users, this can be easily expressed by using refinements for the components in the FIA class that express the specific requirements by using refinements to characterize the type of user for which a specific requirement applies and then use multiple instantiations of components to fully cover all the different types of users. In a similar way different requirements for managing different types of users, subjects, objects or security attributes can quite often be expressed using refinements.

ISO/IEC 15408-1 provides some examples for refinements and how to use them to express requirements more precisely.

Some guidance on how to define extended components can also be found in ISO/IEC 15408-1. The following sections extend this guidance.

Before defining an extended component one should investigate in published and evaluated Security Targets or Protection Profiles if someone else has already defined an extended component one could use to specify the security functional or security assurance requirement one wants to include. Taking an already defined extended component from an evaluated Security Target or Protection Profile has the advantage that the component itself has already been checked for consistency and conformance against the requirements of ISO/IEC 15408 as part of the evaluation of the Security Target or Protection Profile that contained it.

When defining extended requirements, ISO/IEC 15408-1 requires that they are defined in a similar way as existing components in ISO/IEC 15408. This applies to the naming of the extended component, the way they are expressed and the level of detail. It is therefore advisable to describe an extended component using the same structure as is used in ISO/IEC 15408. Concerning the naming of en extended component, one should try to identify if the component fits into one of the classes or even one of the families that are already defined in ISO/IEC 15408 and construct the name by using the class name and (potentially) family name and just add an indicator showing that this is an extended component. Whenever possible the component should be defined in a generic way allowing assignment and/or selection operations. This makes it easier for another ST or PP writer to pick up the extended component and instantiate it in a way that fits their requirements.

Specifying an extended SFR component using ISO/IEC 15408-2 functional components as a model for presentation will involve:

a) defining the extended SFR at a similar level of abstraction as ISO/IEC 15408-2 components;

b) using a similar style and phraseology to ISO/IEC 15408-2 components;

c) using the topology and nomenclature approach for components as in ISO/IEC 15408-2.

Knowing that a new SFR is of a similar nature to others in an existing class or family helps bound its degree of newness and also may help with specific wording for common concepts that occur throughout that class or family.

Particular characteristics of the style of presentation of functional components in ISO/IEC 15408-2 include:

a) most functional requirements begin with the phrase *The TSF shall* or *The TSF shall be able to*, followed by a verb such as *allow, detect, enforce, ensure, limit, monitor, permit, prevent, protect, provide* or *restrict*;

b) the use of standard terms such as *security attribute* or *authorised user*;

c) each element tends to stand on its own and can be understood without reference to previous elements;

d) each security requirement must be evaluatable, i.e. it must be possible to determine whether the requirement has been met by a TOE.

In constructing an extended component, you should also consider whether the SFR:

a) should incorporate any assignment or selection operations to be completed by the ST or PP author;

b) implies any dependencies on other SFRs which must be included in the PP or ST;

c) describes any events which should be auditable, and if so what information should be recorded for the event;

d) has any implications for security management, e.g. relies on security attributes that need to be managed.

If you believe you have a well-constructed SFR that is not included in ISO/IEC 15408-2, and is significantly different from, and would significantly enhance, the existing set of functional components in ISO/IEC 15408, you are advised to submit the SFR for inclusion in the next iteration of that International Standard.

It should be noted that it may not be necessary to specify ISO/IEC 15408 operations such as assignment or selection for SFRs constructed in this way if the SFR is only intended for use in the ST, i.e. there is no intent to reuse the component in other PPs, STs, or functional packages.

Naming for an extended SFR not included in ISO/IEC 15408-2 should use the topology and naming conventions of ISO/IEC 15408-2, to be in the same style as the standard. Extended security functional components should use ‘F’ for function, followed by the appropriate class, and family designations followed by a component number. An extended component based on the existing classes can then be inserted at the appropriate place. Where an extended component is unrelated to existing classes it is acceptable for naming to make it clear that the extended security requirement is new by, for example making the class of the component ‘EX’, or appending ‘EX’ to the end of the component name. How the extended component is denoted should be explained in the application notes for the PP or ST. Care should be taken that the naming convention used does not conflict with ISO/IEC 15408-2.

Annex A provides an example for an extended component and explains it in a way similar as those components defined in ISO/IEC 15408-2. This allows en evaluator to treat the extended component similar to those defined in ISO/IEC 15408-2 when evaluating the Security Target or Protection Profile that defines the extended component.

In a similar way as described in the example in annex A for an extended security functional component one can also define an extended assurance component. This makes sense when a specific assurance activity is common for the type of product described by the Security Target or Protection Profile where this assurance activity is not covered by the existing components in ISO/IEC 15408-3. In addition to the definition of the assurance component in a style similar to that used in ISO/IEC 15408-3, an extended assurance component also requires the definition of an evaluation methodology that explains the activities an evaluator has to perform to verify that a product conforms to the extended assurance component. Those activities have to be defined using the structure and level of detail as defined in ISO/IEC 18045 for the assurance components defined in ISO/IEC 15408-3.

Extended assurance components should provide a definition of the following elements (see ISO/IEC 15408-1, C.5 for more details):

a) developer actions;

b) requirements for the content and presentation of evidence that a developer must provide;

c) evaluator actions.

Inspection of ISO/IEC 15408-3 shows that the elements associated with an assurance component are characterised as follows:

a) developer action elements are intended to express the activities the developer must perform, generally the providing of evaluation evidence;

b) content and presentation elements are intended to characterise the required content and “qualitative” aspects of the evaluation evidence a developer must provide;

c) evaluator action elements take two forms:

  - the first evaluator action is generally of the form:

    The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

  - any further evaluator action elements generally take the form of a statement for independent work and determination on the part of an evaluator.

Therefore, all requirements for content and presentation of evidence should not only be clearly and unambiguously expressed, but also should avoid (as far as possible) requiring subjective judgement on the part of the evaluator. Rather, the extended SAR should define clear objective criteria against which an evaluator may reach a verdict. You should consider providing application notes for any clarification of the extended SAR that is needed in support of the requirement for objective judgement.

To ensure that the extended SARs are specified in the same style as ISO/IEC 15408-3 components, you should ensure that each separable requirement is stated as an individual requirements element. You should also, when choosing the wording of the extended SAR, consult ISO/IEC 15408-1, clause 3 which gives a definition of general English terms that are used in a precise way within ISO/IEC 15408-3.

If you believe you have a well-constructed extended SAR that is not included in ISO/IEC 15408-3, and is significantly different from, and would significantly enhance, the existing set of assurance components in ISO/IEC 15408, you are advised to submit the SAR for inclusion in the next iteration of that International Standard.

When you have defined the extended assurance component, you also need to define the evaluator work units required to show compliance to the extended assurance component in an evaluation. This should be done using the structure of the work units in ISO/IEC 18045 as an example. The work units shall address all aspects of the extended assurance component and give clear advise to an evaluator how to perform the assessment.

12 Specifying the security requirements

12.1 Introduction

This clause provides guidance on the specification of IT security requirements in a PP or ST. This guidance applies the TOE security requirements.

The following types of IT security requirements are specified in a PP or ST:

a) Security Functional Requirements (SFRs) on the TOE. These identify the requirements for security functions which the TOE must provide to ensure that the security objectives for the TOE are achieved.

b) Security Assurance Requirements (SARs) on the TOE. These identify the required level of assurance in the implementation of the SFRs.

This is illustrated in Figure 3 following.

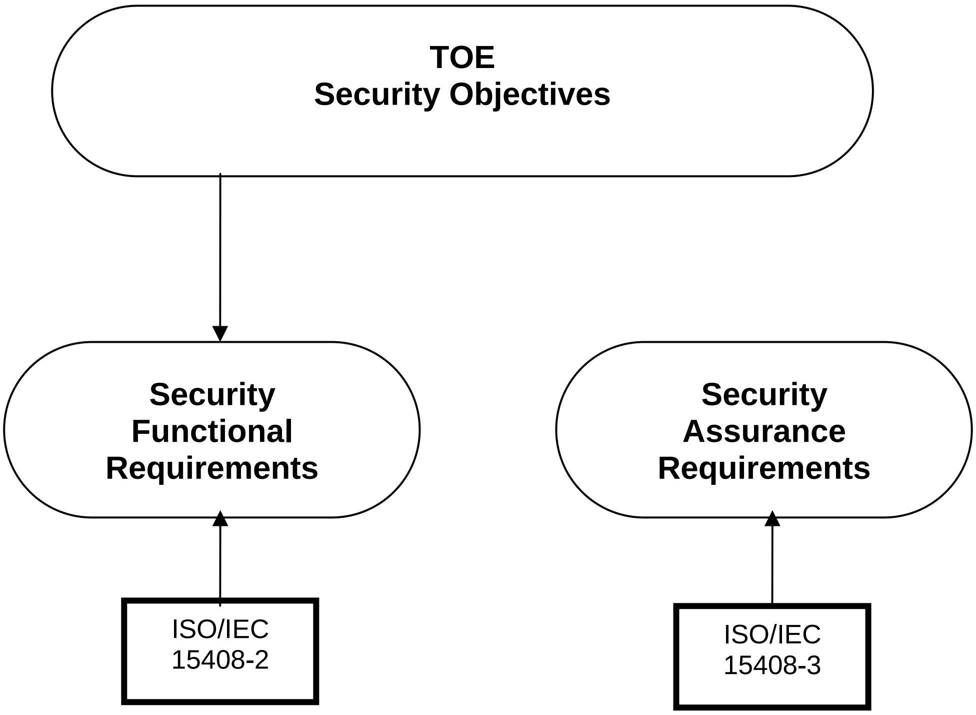
TOE
Security Objectives
Security Functional Requirements
Security Assurance Requirements
ISO/IEC 15408-2
ISO/IEC 15408-3

Figure 3 — Derivation of IT security requirements

As Figure 3 shows, a significant characteristic of the IT security requirements is that they are intended to be constructed, where possible, using the catalogue of functional components defined in ISO/IEC 15408-2 and the catalogue of assurance components defined in ISO/IEC 15408-3, as appropriate. The intent of ISO/IEC 15408 here is to ensure a degree of standardisation in the way the IT security requirements are presented. The use of this 'common language' for expressing IT security requirements is thus intended to facilitate comparison between PPs and STs. A guide how to derive security functional requirements using the functional paradigms of ISO/IEC 15408 is provided in 12.2.

However, ISO/IEC 15408 recognises that there may be cases where there is no appropriate functional or assurance component in ISO/IEC 15408-2 or ISO/IEC 15408-3. In this case, the IT security requirements may be stated explicitly without reference to ISO/IEC 15408; however, such IT security requirements must be unambiguous, evaluatable, and expressed in a similar style to existing ISO/IEC 15408 components. 12.3.7 provides guidance where no appropriate functional components can be identified in ISO/IEC 15408-2; 12.4.3 provides similar guidance in respect of assurance components.

ISO/IEC 15408 permits a degree of flexibility in the way the SFRs and SARs are specified by allowing a set of *operations* to be performed on them to tailor the security requirement appropriately - namely assignment, iteration, selection and refinement. 12.3.2 below provides guidance on the use of operations on ISO/IEC 15408 functional components. 12.4.2 does the same for ISO/IEC 15408 assurance components.

Each security component in ISO/IEC 15408-2 and ISO/IEC 15408-3 is assigned its own unique reference in ISO/IEC 15408, based on a defined taxonomy:

a) in ISO/IEC 15408-2, for example, component FAU_GEN.1.2 has the following meaning:

- 'F' indicates it is a *functional* requirement;

- ‘AU’ indicates it belongs to the *security audit* class of SFRs;
- ‘GEN’ indicates it belongs to the *security audit data generation* family within that class;
- ‘1’ indicates it is the *audit data generation* component within that family;
- ‘2’ indicates it is the second *element* within that component.

b) the components in ISO/IEC 15408-3 use a similar taxonomy, but additionally identifies each *element* as belonging to one of three sets of *assurance elements*, by appending a letter:

- the letter ‘D’ indicates it belongs to the set of *developer action elements,* the activities performed by the developer;
- the letter ‘C’ indicates it belongs to the set of *content and presentation elements,* the information the evidence is meant to convey;
- the letter ‘E’ indicates it belongs to the set of *evaluator action elements*, the activities performed by the evaluator.

c) in ISO/IEC 15408-3, for example, component ADV_TDS.1.2C has the following meaning:

- ‘A’ indicates it is an *assurance* requirement;
- ‘DV’ indicates it belongs to the *development* class of SARs;
- ‘TDS’ indicates it belongs to the *TOE design* family within that class;
- ‘1’ indicates it is the *basic design* component within that family;
- ‘2’ indicates it is the second *element* in a set of *assurance elements*;
- ‘C’ indicates it is an *element* in the set of *content and presentation elements* within that component.

SFRs and SARs are selected at the *component* level: all defined elements within that component have to be included in the PP or ST if the component is to be included. There are two types of *relationships* between components which you need to be aware of, as these have a bearing on the process of selecting the IT security requirements:

a) Components within a family may have a *hierarchic* relationship, indicating that one component includes all requirements specified in another component in that family. For example, FAU_STG.4 is hierarchic to FAU_STG.3 because all functional elements defined in the latter are also included in the former. However, FAU_STG.4 is not hierarchic to FAU_STG.1, and it is therefore possible to include both components in the same PP or ST.

b) Components may have defined *dependencies* on any component in any other family indicating that when a component is not self sufficient and relies upon the functionality of, or interaction with, another component for its own proper functioning. For example, FIA_UAU.1 (which requires authentication of any user’s claimed identity) has a dependency on FIA_UID.1 (which requires users to be identified). These components must also be included in a PP or ST, unless the dependencies can be shown not to be relevant to the threats and security objectives.

12.2 The security paradigms in ISO/IEC 15408

12.2.1 Explanation of the security paradigms and their usage for modelling the security functionality

To provide a better understanding of the structure of the classes, families and components defined for the security functional requirements in ISO/IEC 15408-2, this guide extends on the security functional paradigms expressed in ISO/IEC 15408-2, Clause 5.

The purpose of the security paradigms in ISO/IEC 15408 is to provide a basis for modelling the security functions of a TOE to the extent required to show that the security objectives can be met in that model. The paradigms explained in the previous sections can now be used to develop an abstract model of the security functions, which is then expressed using the SFRs defined in ISO/IEC 15408-2. The following sections provide guidance how to develop such a model and describe it using the SFRs.

12.2.2 Controlling access to and use of resources and objects

12.2.2.1 Explanation

In the paradigm of ISO/IEC 15408-2 security functions control and regulate the use of resources protected by the TOE. Resources may either be internal to the TOE (like main memory, CPU time, disk space, services etc.) or may itself be outside of the TOE but only accessible (at least for some entities) under the control of TOE functions (e. g. network services from other systems). A firewall is a typical example of a TOE that controls use of resources that itself are not part of the TOE.

Examples of resources that may need to be controlled to achieve the security objectives are:

- Memory (both main memory and disk space)

- CPU time

- Peripheral devices or network links

- Functions

Users are defined in ISO/IEC 15408-1 as "any entity (human or IT) outside the TOE that interacts (or may interact) with the TOE". Subjects are defined in ISO/IEC 15408-1 as "an active entity in the TOE that performs operations on objects". Users and subjects are the active entities that request services from the TOE and thereby operate on objects and resources.

In order to achieve its security objectives, the use of resources is regulated within the TOE based on rules that the TOE needs to enforce. Those rules may regulate the use of resources as well as record the use of resources.

A - deliberately incomplete - list of parameters that may be evaluated as part of such rules are:

- The type and identity of the entity that initiated the request

- Other attributes of the entity that initiated the request

- The type and identity of the resource that is targeted by the request

- Other attributes of the resource that is targeted by the request

- The type of request

- The time and date

- The internal state of the TOE

To enforce rules based on those parameters, the TOE needs maintain and manage those parameters:

- For external entities (also called "users") it needs to identify and potentially also authenticate the external entity – at least to the extent required to enforce the rules. If the rules are just based on the external entity belonging to a specific set or group of external entities, it is sufficient for the TOE to identify (and potentially authenticate) the set or group.

- Quite often the TOE maintains a list of external entities (potentially with their security attributes) that are allowed to use services controlled by the TSF. In this case functions are required to manage the list of external entities and their security attributes (provided the list is not static).

The part of a TOE that implements the security functions used to satisfy the security objectives as well as all other parts of the TOE that have the potential to modify or bypass the security functions is called the "TOE Security Functionality" (TSF). Depending on the architecture of a TOE the TSF may be the whole TOE or may be a defined part of the TOE. If the TSF are just a part of a TOE it is important that the non-TSF parts of the TOE can not manipulate or bypass the TSF in a way that violates the security objectives.

Both external entities as well as subjects that request services using controlled resources will use the interfaces to the TSF, called the TSFI.

In some cases subjects will operate on behalf of external entities. In those cases the external entity (or user) "binds" to the subject. As part of this binding process, the security attributes of the subject will often be modified to reflect this binding. An example are TOEs where the subject inherits the security attributes of the external entity, but more complex rules may exist defining how the security attributes of the subject are derived as part of the binding process.

Resources may be grouped to "objects" and the TOE may have dedicated rules for using those objects that are different from the rules for using the resources that make up the object. A typical example is a TOE that enforces a rule for maximum quotas for disk space (the resource) and rules that control access to the files (the objects) that are constructed from the disk space resources. This example shows that a single resource may be subject to different rules enforced by the TOE where one set of rules regulates the use of the resource and another set of rules for the objects constructed from the resources.

The rules regulating access to and use of objects are usually different for different types of objects. To avoid confusion ISO/IEC 15408 allows grouping the set of rules for different objects, subjects and operations into different "Security Function Policies" (SFPs) and referencing the SFP in the individual SFRs to indicate the security function policy to which the SFRs belong. A security function policy always needs to have a defined scope, which is the definition of the set of subjects, users, objects, resources and operations to which the policy applies. This definition must be unambiguous to ensure that the scope of the SFP is well defined. Then the rules enforced by the operations for the subjects or users when using the objects or resources are defined as part of the SFP. As mentioned above those rules usually will be based on specific attributes of the subjects, users, objects or resources. Those attributes that influence the rules of the SFP are called "security attributes". The requirements for management of the security attributes that play a role in a SFP are also part of the SFP, including the definition how the security attributes are initialized when an entity subject to the SFP is created, imported or registered (for users). To summarize, a SFP describes the rules for access to and use of a defined set of objects or resources by a defined set of active entities (users or subjects) using a defined set of operations together with the functions to manage the security attributes used in those rules.

A typical example is an access control policy for file system objects in an operating system. The active entities are processes, some of which operate on behalf of a user and therefore have security attributes derived from the user security attributes upon binding. The operations are those system calls that operate on file system objects like opening a file for read, write or update, view or change the attributes of a file, creating or deleting a file. In addition there are operations that manage the security attributes of the processes or the file system objects. Typical examples of security attributes that may play a role in such a SFP are:

- object security attributes: access control lists, file type
- user security attributes: user identities, user roles
- process security attributes: process identity, process trust level

Other SFPs may regulate operations external entities perform directly without an intermediate subject. An example is a firewall system that regulates how the network services and functions can be used by an external system. Still there are active entities (external systems that initiate the request), objects (external systems that are target of the request) and operations (network services). The rules of such an SFP may be based on the identity of the external systems involved in the operation, the type of operation performed (e. g. the port used), the context of the operation (e. g. if a connection on a specific port has been established previously) and/or the content of network packages.

It is not unusual to define more than one SFP even for the same set of users, subjects, objects and operations. An example is a discretionary access control policy as one SFP and a mandatory access control policy as an additional SFP. Although the set of users, subjects, objects and operations addressed by the SFP are the same, the rules of the SFP and the set of security attributes used in those rules are different and justify defining two SFPs.

12.2.2.2 Usage

Access control policies provide a basis to model the TOE in terms of resources and objects as well as operations allowed on those resources and objects by the TOE (or via the TOE) to active entities (either inside or outside of the TOE). So the first step when deriving a model of a TOE usable to specify security functional requirements for access control is to identify the resources, objects, operations provided by the TOE as well as the subjects and users that trigger the operations. In an initial step the model should only include those types of resources, objects, operations, subjects and users in the model that can be directly derived from the security objectives and the general TOE functionality described in the beginning of the PP or ST. When developing a ST for an existing product or system, the entities defined in the model should exist in the TOE. Additions to this first sets may be required when defining the security functional requirements to ensure consistency and completeness.

Defining entities in the model that do not exist in the TOE will lead to problems during the evaluation since ISO/IEC 15408 assumes that the SFRs and the entities mentioned in the SFRs are abstractions of entities that exist in the TOE and can therefore be mapped by refinement to entities in the design and implementation of the TOE.

In the next step, rules need to be defined that regulate access and use of the resources and objects via the operations for the subjects and/or users defined in the model such that the security objectives are satisfied. Again, when defining a ST for an existing TOE, the rules shall of course attempt to be an abstraction of the real behaviour of the TOE for the entities defined in the model such that the rules implemented by the TOE are strict refinements of the rules in the model.

Part of the definition of the rules is the identification of the parameters that are used in those rules. Most likely you have to define "security attributes" of the resources, users, subjects, and objects. Those security attributes should be collected in a list, since rules for the initialization and management may be required.

When defining those rules you will quite often identify that rules differ for different set of resources, objects, users, subjects or operations. To simplify the description of the model, you should group sets ("types") of resources, objects, users, subjects, and operations with identical (or almost identical rules) into security function policies. Give each security function policy a name that identifies it.

Define the rules for creating and deleting subjects and objects. Those rules may be different for different types of subjects and objects. They also need to define how the security attributes of the subjects and objects are initialized.

Define the rules for the management of the security attributes of subjects and objects in cases where those attributes are not static. Note that those rules may involve operations triggered by external entities through the TSFI as well as rules that describe how security attributes are modified as part of operations performed by the TSF.

Define the rules for registering ("creating") and de-registering ("deleting") users when users need to be registered to the TOE. Rules for user registration also include the rules for the initialization of user security attributes. Note that there are cases where users do not need to be registered. They can request services and identify and potentially authenticate themselves using credentials they present. Those credentials may also include security attributes of the user. In those cases rules need to be defined that define the credentials accepted and how the credentials are checked.

Define the rules for identification and (if required) authentication of users. Those rules define the credentials the user has to present (type of credentials, potential restrictions on those credentials like minimum and maximum length, minimum and maximum lifetime etc.) as well as the reaction of the TSF when incorrect credentials are presented.

Define the rules for the management of the security attributes of users. This is done in a similar way as defining the security attributes of subjects and objects.

If the TOE supports a function of user-subject binding, define the rules involved in this binding. Those rules may include:

- conditions that need to be satisfied to allow the binding

- setting of the security attributes of the subject after the binding

When this is done one has to review if additional management rules are required. An example for such an additional rule is one that allows creating a new security attribute (e. g. a new user role) potentially together with rules that define how to manage this security attribute (e. g. define the set of user security attributes a user gets as part of the role).

12.2.3 User management

12.2.3.1 Explanation

In the paradigm of ISO/IEC 15408 a user is an entity external to the TOE that requests services from a TOE using its interfaces. Users may need to be "registered" before they can use TOE services or the TOE may allow users to request services without being previously registered. In many cases the decision of the TOE whether to provide the requested service depends on some security attributes of the user. User security attributes may either be submitted by the user together with the request or may be derived from data the TOE has stored about the user or the group the user belongs to.

In the first case the TOE needs to ensure that the security attributes submitted by the user can be trusted. This implies that the TOE implements rules how to evaluate the security attributes and establish trust that the user (which may be unknown) uses the security attributes legitimately.

In the second case the TOE needs to know the identity of the user or the identity of the group the user belongs to. Also in this case the TOE needs to implement rules specifying how to verify that the claimed identity of the user or the user's membership in the group is correct. This process is called authentication and requires that the user presents credentials used by the TOE to establish its trust in the correctness of the identity or group membership claimed. Rules need to be defined that specify how the authentication process is performed and how the parameter of the authentication process can be managed.

When users are required to be registered, there is a need to define the rules how users can be registered and how their security attributes can be managed.

In some cases the TOE will use one of its subjects to act on behalf of a user. In this case the subject is "bound to the user" by the TSF, i. e. the TSF will have rules that define how a subject's security attributes are derived when the subject is bound to a user. Very often the subject inherits part of the security attributes of the user allowing to enforce user security attribute based access control policies even when the actual access is performed by a subject.

12.2.3.2 Usage

To define the functions for user management you need to perform the following steps:

- identify and define the types of users that can access the TOE (together with the set of security attributes that each type of user may have).

- identify for each type of user if he needs to be registered before using TOE functions.

- for each type of user that needs to be registered, define the rules for user registration (how this is done) and the security attributes of the user that need to be set upon registration.

- identify for all type of users, if user identification is required. If yes, define the rules how a user is identified.

- identify for all type of users, if user authentication is required. If yes, define the rules how a user is authenticated. Define the conditions under which a user needs to be authenticated.

- define the rules how the authentication process can be managed (including the management of credentials used for authentication).

- for each type of users define the rules how user security attributes can be managed.

- when user-subject binding is possible or required, define the rules for this binding. Especially define the rules how the subject's security attributes are set during the binding process.

12.2.4 TOE self protection

12.2.4.1 Explanation

Protecting the security functions itself is required whenever one of the following conditions holds:

- there is a possibility for a threat agent to attack the security functions within the intended environment of the TOE such that a security objective can not be achieved.

- there is a possibility that a security objective can not be achieved due to a malfunction of an element of the TOE environment.

- there is a possibility that a security objective can not be achieved due to a malfunction of an element of the TSF.

In those cases self-protection functions as part of the TSF need to be defined, that detect and react on those conditions in a way which achieves the security objectives also in those conditions.

Defining TOE self protection in the functional model requires:

- identification of possible attack scenarios and malfunctions that may violate a security objective

- identification of a function that is able to prevent the attack or malfunction. An example for a such a function is an increased physical protection of the TOE that prevents specific physical attacks.

- in cases where prevention is not possible (which usually is the majority), identification of functions that detect the attack or malfunction and react properly.

Detection of an external attack or a malfunction of a system in the TOE environment may require monitoring the use of TSFI and checking for conditions that result from an attack, monitor conditions on communication links that result from an attack or monitor sensors the TOE has specifically to detect attacks.

12.2.4.2 Usage

To define the TOE self protection functions you need to identify from the security problem definition if such functions are required to satisfy the security objectives. When this is the case, you need to select if you need to prevent an external attack (e. g. by some enhanced physical protection) or if you need to detect an attack or a malfunction and react to it.

You start with a list of attacks or malfunctions that may occur in the intended environment of the TOE which, when not dealt with, potentially violate the security objectives. For each list one should define how the attack or malfunction is intended to be handled, i. e. if it is prevented by a TOE security functionality implemented by the TOE or if TOE security functionality for the TOE needs to be defined that detect the attack or malfunction and react to it.

In the case of a function preventing an attack, the function needs to be described with some justification which types of attack it is supposed to counter.

In the case of detection and reaction the criteria and rules for detection (on an abstract level) and the reaction need to be defined (as abstract rules stating what the TOE is supposed to do in such a case).

Detection of malfunctions of the TSF may be done by monitoring internal state variables, internal functions performing tests or by having functions or data redundant and check for inconsistencies.

A reaction may result in:

- a corrective action that eliminates the effect of the attack or malfunction. Examples are functions that can detect and automatically correct failures based on redundancy in the data or functionality.

- a corrective action that partly eliminates the effect of the attack or malfunction but results in some reduction of the functionality of the TOE (which needs to be consistent with the security objectives). Examples are functions recover from a failure or attack, but recovery may take time and may not be complete. In those cases it needs to be ensured that neither the delay nor the loss in functionality or data resulting from an incomplete recovery violates any security objective.

- preparing the TOE for manual corrective action (e. g. stopping the parts of the TOE that are affected by the attack or malfunction or the whole TOE, requiring the stopped parts or the whole TOE to be restarted in a secure mode).

- stopping the failed parts of the TOE or the whole TOE without providing a method within the TSF to restart securely. An example is a TOE that destroys important functions or data when detecting an attack or malfunction to ensure that the TOE does not violate its security objectives.

The list of corrective actions above is sorted with increasing impact on the overall functionality of the TOE.

12.2.5 Securing communication

12.2.5.1 Explanation

Functions that protect data when communicating either with an external entity or when communicating between different parts of a distributed TOE using an unreliable or untrusted communication channel are another example of functions that require additional modelling. To model communication, the security properties of the communication channel need to be defined. Such properties may include:

- authentication of communication partners
- integrity protection of data transferred over the channel (which may include protection against replay of messages and/or changing the sequence of messages)
- confidentiality protection of data transferred over the channel
- protection against loss of data
- providing non-repudiation of send and/or receipt of messages

To model a communication channel the peers of the communication as well as the security properties of the channel need to be defined. This applies to both on-line as well as off-line communication channels.

12.2.5.2 Usage

Identification of functions required to secure communication requires the following steps:

- identification of communication links
- definition of security properties required for each communication link. Examples of such security properties are:
  - authentication of communication peers
  - integrity protection (potentially including replay protection, message sequence protection etc.)
  - confidentiality protection (potentially including protection against traffic flow analysis)
  - provision of non-repudiation (for sending, receipt, or both)
  - provision against loss of communication data

For each communication link the security properties required need to be defined. In an ST also the mechanisms used to implement those security properties are defined (especially cryptographic mechanisms). In a PP the mechanisms should be defined only up to the level of detail required. Note that this level of detail may be quite high when any TOE compliant to the PP is also supposed to satisfy interoperability requirements. In those cases even a PP may specify the mechanism down to the level of a specific protocol together with protocol options (e. g. cryptographic algorithms) that are required to ensure interoperability.

When identifying the list of communication links you should not only look for physical communication links but also identify logical links (e. g. on an application protocol level) that require specific protection. Such communication links may well be stacked at different protocol levels where the individual levels provide different types of protection. For example IPsec on the IP level may provide the authentication of peer entities (in this case the systems) as well as integrity and confidentiality protection. An application protocol (which may represent a different logical communication link) on top of IPsec may then provide additional authentication (e. g. of the human user or the application) as well as non-repudiation functions. In this case IPsec and the application protocol should be listed as different communication links with their own specific security properties.

Note that most of functions for securing communication links enforce integrity protection and protection against loss of data by functions that detect those conditions. Similar to detection functions described in the section on TOE self protection, the reaction of the TOE when those conditions are detected may need to be defined. Also the reaction on failed authentication attempts and invalid non-repudiation may need to be defined.

Note that exporting TSF or user data from the control of the TOE and importing TSF or user data into the TOE can be considered as a special case of communication where the communication peer is unknown. In the case of export and import the following properties may be considered:

- integrity protection (potentially including replay protection, freshness etc.)
- confidentiality protection
- provision of non-repudiation (for export, import, or both)

12.2.6 Security audit

12.2.6.1 Explanation

Monitoring defined security critical events and maintaining records of those events for future analysis or for evaluation in automated responses to such events is another security function that may be required for a TOE to satisfy the security objectives. Security critical events may be those directly related to requests to use TOE services by an active entity as well as the detection of a security critical state or event that can not be directly related to such a request.

Examples of security critical events are:

- successful and/or rejected attempts to use services provided by the TSF
- unexpectedly reaching a failure state
- unexpected or faulty behaviour of a remote trusted IT product
- failure detected by a self-test function
- exceeding defined security critical thresholds
- changes to critical TSF data
- accumulation of events where each individual event is not considered critical enough to be audited

12.2.6.2 Usage

In order to model security audit it is required to:

- list the events that need to be audited
- define the rules regulating when the event is audited (e. g. only when a request is denied)
- define the data that needs to be collected for each event
- define the rules how the collected audit data is processed and analyzed

It is good practice to analyse for each individual security functionality if there are events associated with this functionality that need to be audited. In addition the model of the security functions should be analyzed for critical internal states that need to generate an audit record when reached.

12.2.7 Architectural requirements

12.2.7.1 Explanation

In addition to the requirements listed above there may be a need to specify requirements for the architecture of the TOE. Such requirements may be needed in order to ensure that it is possible to perform an analysis of the architecture as well as supporting reader's' understanding of the TOE's architecture. They usually are related to specific properties the TOE shall enforce. Typical examples of such properties are:

- fault tolerance
- information flow control
- privacy properties
- real-time properties

Architectural requirements are often supported by requirements from the previous sections. For example information flow control and privacy properties are usually accompanied by specific rules regulating access to objects and fault tolerance is usually accompanied by security audit requirements used to detect a fault. Those access control rules, especially security audit rules, are necessary, but usually not sufficient to enforce the property requirement.

Architectural requirements are more difficult to identify and specify than the other security functional requirements. Nevertheless they may be required to completely meet some security objectives and they therefore need to be defined as part of the security functional requirements in a PP or ST.

12.2.7.2 Usage

To identify and model architectural requirements is done using the following steps:

- identify the security objectives that have not been addressed or not been fully addressed by requirements identified in the previous steps
- identify the architectural support required to satisfy those objectives
- define rules that contribute to this architectural support

Little help can be provided in this guide how to select architectural requirements. In the case of an ST, those requirements will most likely be predefined by the architecture of the TOE the ST is developed for. For example if the TOE is known to be distributed, requirements for keeping data consistency between distributed parts of the TOE or requirements to protect data from unauthorized access when transmitted between distributed parts of the TOE may be required in order to achieve defined security objectives. Although one may argue that supporting internal functions of the TSF within TOE should be redundant as long as the TOE meets its security objectives at its TSFI, specifying mandatory internal functions that support the security objectives helps in understanding and analyzing a TOE during an evaluation.

12.3 How to specify security functional requirements in a PP or ST

12.3.1 How should security functional requirements be selected?

Having defined the security objectives for the TOE as part of the security problem definition, you now need to elaborate on how these security objectives are to be met. This is done by selecting an appropriate set of SFRs which, as stated above, is done at the *component* level. Of course, the SFR selection process will be significantly easier if pre-defined functional packages are available that are relevant to the security objectives for the TOE.

The SFRs are selected based on a model of the overall functionality of the TOE. This functional model defines resources, users, subjects, objects and operations. The SFRs then define the security functionality such that the security objectives are met within the functional model of the TOE. As with any model it is an abstraction of the real functionality of the TOE but the level of abstraction should be sufficient to understand the principle functions of the TOE. Resources, users, subjects, objects and operations that do not need to be controlled to met the security objectives can be neglected when defining the SFRs. For example if the only security objective of a TOE is to control access to data, the resource "CPU time" may not need to be considered when defining the SFRs.

There are several stages to the process of selecting the SFRs for a PP or ST. In considering the selection process, it is helpful to distinguish between the following two types of SFR:

a) *principal* SFRs, which *directly* satisfy the identified security objectives for the TOE;

b) *supporting* SFRs, which do not *directly* satisfy the security objectives for the TOE, but which nonetheless provide support to the *principal* SFRs, and hence *indirectly* help satisfy the relevant security objectives for the TOE.

Whilst ISO/IEC 15408 does not *explicitly* distinguish between these two types of SFRs, such a distinction is *implicit* in the consideration of such things as dependencies between functional components, and the demonstration of mutual support between SFRs. Therefore, whilst there is no need for you to explicitly categorise the SFRs as *principal* or *supporting* in the PP or ST, recognising that there are these two types of SFR will be of significant benefit when you come to write the PP or ST Rationale.

The first stage in the SFR selection process is thus, for each security objective for the TOE, to identify the *principal* SFRs for the functional model which directly satisfy them. Once a complete set of *principal* SFRs has been established, there then follows an iterative process whereby the complete set of *supporting* SFRs are identified. As described above, all SFRs (whether *principal* or *supporting*) should, where possible, be expressed using appropriate functional components from ISO/IEC 15408-2. Subclause 12.3.2 provides guidance identifying which functional components should be used to express common security functional requirements. When selecting functional components from ISO/IEC 15408-2, you should also consult the guidance contained in the annexes to ISO/IEC 15408-2 as to whether the component would be appropriate, and how it should be interpreted.

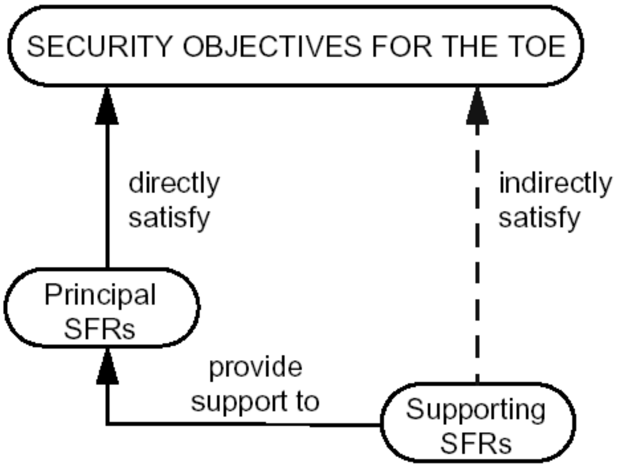

Figure 4 — Role of principal and supporting SFRs

The relationship between these two types of SFR is illustrated in Figure 4 above. It may be noted that this relationship is relevant to the PP or ST rationale, which, *inter alia*, is required to demonstrate mutual support between the SFRs. This will involve providing an explanation of the nature of the support provided by *supporting* SFRs in helping to ensure that the security objectives for the TOE are met.

There are three stages involved in identifying the complete set of *supporting* SFRs:

a) Identify the additional SFRs needed to satisfy (where you consider it appropriate) the dependencies (as defined in ISO/IEC 15408-2 for the relevant functional components) of all *principal* SFRs. This includes any dependencies of the *supporting* SFRs identified during this stage.

b) Identify any additional SFRs that are necessary to ensure that the security objectives for the TOE are achieved. This will include SFRs needed to defend the *principal SFRs* against composite attacks that first defeat the function, then mount the threat the function is intended to counter.

c) Identify the additional SFRs needed to satisfy (where you consider it appropriate) the dependencies of those *supporting* SFRs selected during the second and third stages.

The identification of *supporting* SFRs to satisfy the dependencies as identified in ISO/IEC 15408-2 is likely to be an iterative process, for example:

a) Suppose that the PP or ST includes a security objective requiring the TOE to provide specific responses to the detection of events indicative of an imminent security violation. This leads to the inclusion of a *principal* SFR based on the FAU_ARP.1 (Security Alarms) component.

b) According to ISO/IEC 15408-2, FAU_ARP.1 has a dependency on FAU_SAA.1 (Potential Violation Analysis) which should also be included as a *supporting* SFR.

c) FAU_SAA.1 has a dependency on FAU_GEN.1 (Audit Data Generation).

d) FAU_GEN.1 has a dependency on FPT_STM.1 (Reliable Time Stamps).

e) FPT_STM.1 introduces no requirements for additional functional components.

It should be noted that ISO/IEC 15408 permits you to leave some dependencies ‘unsatisfied’, provided you explain why the relevant SFRs are not required to satisfy the security objectives (and hence address the security concerns).

Dependencies should be applied in a consistent manner. For example, in the case of FAU_ARP.1, consistency is ensured by the nature of the requirements (FAU_ARP.1 depends on the expectation of a potential security violation that is defined by application of FAU_SAA.1.2).

For other components, consistency may be more problematic. For example, in the case of FDP_ACC.1, the PP or ST will identify the particular access control SFP to which it relates. In satisfying the dependency of FDP_ACC.1 on FDP_ACF.1, it must be ensured that FDP_ACF.1 is applied to the same access control SFP that was used for FDP_ACC.1. If the iteration operation is applied to FDP_ACC.1 for different access control SFPs, the dependency on FDP_ACF.1 will need to be satisfied in respect of each such access control SFP.

The identification of additional *supporting* SFRs (i.e. those that are not identified as dependencies in ISO/IEC 15408-2) involves identifying any other SFRs which you consider to be necessary to support the achievement of the security objectives for the TOE. Such SFRs will typically provide support by reducing the options or opportunities available to an attacker, or by increasing the level of expertise or resources an attacker must have to mount a successful attack. The following should be considered in the light of the security concerns and the security objectives:

a) SFRs based on relevant components from the same class in ISO/IEC 15408-2. For example if the component FAU_GEN.1 (Audit Data Generation) is included then this may imply a need to create and maintain a secure audit trail to store the data generated (requiring one or more functional components from the FAU_STG family) and a need for tools to review the generated audit data (requiring one or more functional components from the FAU_SAR family). Alternatively, the generated data may be exported to another system for review.

b) SFRs based on relevant components from the FPT (Protection of the TOE Security Functions) class. Such SFRs will typically protect the integrity and/or availability of the TSF or TSF data on which the other SFRs rely, although they may protect its confidentiality as well. Examples include FPT_TEE.1 (Testing of External Entities) and components from the FPT_PHP (Physical Protection) family, which may be required to support the security objectives where there is an identified need to protect the TSF against such things as TSF failure, corruption, or modification (possibly by malicious means).

c) SFRs based on relevant components from the FMT (Security Management) class. These components will be used to specify any necessary supporting security management SFRs. An example of this would be FMT_REV.1 which addresses the revocation of security attributes, and may be considered relevant where SFRs are included that deal with security attributes (e.g. access control).

The selection of these *supporting* SFRs should always be done in light of the security objectives and the functional model, in particular taking into account the need to end up with a set of SFRs which form a mutually supportive and integrated and effective whole. The process of constructing the PP or ST rationale may therefore have a significant influence on this selection process. You are strongly advised to avoid including *supporting* SFRs that are not needed to achieve the security objectives, because this will only serve to limit the acceptability of the PP or ST given that:

a) some TOEs may not be able to meet such SFRs;

b) increasing the number of SFRs will increase the cost and maintenance of unneeded requirements in evaluation.

If the PP or ST is being constructed using a related PP as a basis, the process for selection of SFRs should be simplified considerably. The PP or ST being constructed should include different SFRs, where appropriate, taking into account any differences between the TOE security problem definition and/or security objectives.

12.3.2 Selecting SFRs from ISO/IEC 15408-2

The following tables provides a mapping between the paradigms explained and the SFR components defined in ISO/IEC 15408-2. Some components cover more than just one aspect of the paradigm and are therefore listed more than once in the tables.

Table 1 — Access control

Requirement	Applicable components
Define subjects, objects, operations	FDP_ACC.1, FDP_ACC.2, FDP_IFC.1, FDP_IFC.2, FMT_SMF.1
Define security attributes	FDP_DAU.1, FDP_DAU.2, FDP_IFF.1, FDP_IFF.2, FRU_PRS.1, FRU_PRS.2, FRU_RSA.1, FRU_RSA.2
Create subjects, objects	FDP_ITC.1, FDP_ITC.2, FMT_SMF.1
Export objects	FDP_ETC.1, FDP_ETC.2
Manage security attributes	FDP_ITC.2, FIA_USB.1, FMT_MSA.1, FMT_MSA.2, FMT_MSA.3, FMT_MTD.1, FMT_MTD.2, FMT_MTD.3, FMT_REV.1, FMT_REV.2, FMT_SAE.1, FTA_LSA.1
Define rules for access	FDP_ACF.1, FDP_IFF.1, FDP_IFF.2, FDP_ROL.1, FDP_ROL.2,FRU_PRS.1, FRU_PRS.2, FRU_RSA.1, FRU_RSA.2
Manage access control rules	FMT_MOF.1, FMT_SMF.1

Table 2 — User management

Requirement	Applicable components
Define user types	FMT_SMF.1
Define security attributes	FIA_ATD.1
User identification rules	FIA_UID.1, FIA_UID.2
User authentication rules	FIA_AFL.1, FIA_SOS.1, FIA_SOS.2, FIA_UAU.1, FIA_UAU.2, FIA_UAU.3, FIA_UAU.4, FIA_UAU.5, FIA_UAU.6, FIA_UAU.7
Management of user credentials and security attributes	FMT_MSA.1, FMT_MSA.2, FMT_MSA.3, FMT_MSA.4, FMT_MTD.1, FMT_MTD.2, FMT_MTD.3, FMT_REV.1, FMT_REV.2, FMT_SAE.1, FMT_SMR.1, FMT_SMR.2, FMT_SMR.3, FTA_LSA.1, FTA_MCS.1, FTA_MCS.2
Manage identification and authentication rules	FMT_MOF.1, FMT_MTD.1, FMT_MTD.2, FMT_MTD.3, FMT_SMF.1
Management of user-subject binding	FIA_USB.1

Table 3 — TOE self protection

Requirement	Applicable components
Detection of malfunction	FPT_TEE.1, FPT_ITI.2, FPT_ITT.3, FPT_PHP.1, FPT_PHP.2, FPT_PHP.3, FPT_RPL.1, FPT_TST.1, FRU_FLT.1, FRU_FLT.2
Reaction to malfunction	FPT_ITT.3, FPT_PHP.2, FPT_PHP.3, FPT_RCV.1, FPT_RCV.2, FPT_RCV.3, FPT_RCV.4, FPT_RPL.1, FRU_FLT.1, FRU_FLT.2
Manage detection and reaction rules	FMT_MOF.1, FMT_MTD.1, FMT_MTD.2, FMT_MTD.3, FMT_SMF.1

Table 4 — Securing communication

Requirement	Applicable components
Establish communication link	FMT_SMF.1, FTP_ITC.1, FTP_TRP.1
Define communication link properties (security attributes)	FCO_NRO.1, FCO_NRO.2, FCO_NRR.1, FCO_NRR.2, FDP_UTC.1, FDP_UIT.1, FDP_UIT.2, FDP_UIT.3, FPT_ITC.1, FPT_ITI.1, FPT_ITI.2, FPT_RPL.1, FTP_ITC.1, FTP_TRP.1
Manage communication link properties	FMT_MSA.1, FMT_MSA.2, FMT_MSA.3, FMT_MTD.1, FMT-MTD.2, FMT_MTD.3, FMT_REV.1, FMT_REV.2, FMT_SAE.1
Manage link establishment rules	FMT_MOF.1, FMT_MTD.1, FMT_MTD.2, FMT_MTD.3, FMT_SMF.1, FTA_SSL.1, FTA_SSL.2, FTA_SSL.3, FTA_SSL.4, FTA_TAB.1, FTA_TAH.1, FTA_TSE.1

Table 5 — Audit

Requirement	Applicable components
Define events to be audited	FAU_GEN.1, FAU_GEN.2, FAU_SEL.1
Define reaction on events	FAU_ARP.1, FAU_SAA.1, FAU_SAA.2, FAU_SAA.3, FAU_SAA.4
Define management of events	FAU_SAR.1, FAU_SAR.2, FAU_SAR.3

Requirement	Applicable components
Define management of audit trail	FAU_STG.1
Manage audit rules	FMT_MOF.1, FMT_MTD.1, FMT_MTD.2, FMT_MTD.3

Table 6 — Architectural requirements

Requirement	Applicable components
Audit trail protection	FAU_STG.2, FAU_STG.3, FAU_STG.4
Cryptographic functions	FCS_CKM.1, FCS_CKM.2, FCS_CKM.3, FCS_CKM.4, FCS_COP.1
Information flow control	FDP_IFF.3, FDP_IFF.4, FDP_IFF.5, FDP_IFF.6
Internal TOE transfer	FDP_ITT.1, FDP_ITT.2, FDP_ITT.3, FDP_ITT.4
Residual information protection	FDP_RIP.1, FDP_RIP.2
Stored data integrity	FDP_SDI.1, FDP_SDI.2
Management	FMT_MTD.1
Privacy protection	FPR_ANO.1, FPR_ANO.2, FPR_PSE.1, FPR_PSE.2, FPR_PSE.3, FPR_UNL.1, FPR_UNO.1, FPR_UNO.2, FPR_UNO.3, FPR_UNO.4
Fail secure	FPT_FLS.1
Availability	FPT_ITA.1, FPT_ITT.1, FPT_ITT.2
Synchronisation of state	FPT_SSP.1, FPT_SSP.2
Secure time stamp	FPT_STM.1
Data consistency	FPT_TDC.1, FPT_TRC.1

The tables are intended to help identifying suitable SFR components once the security functional model has been defined in accordance with the guidance in subclauses 12.2. and 12.3.1 above. It is left to the author of an ST or PP which component he selects and how he expresses the aspect of the security functional model using the component and the operations allowed.

For the architectural requirements, a list of possible architectural issues is provided, which is mapped to SFR components from ISO/IEC 15408-2 that are related to those issues.

12.3.3 How to perform operations on security functional requirements

12.3.3.1 Permitted operations

As stated in 10.1 above (see also ISO/IEC 15408-2, 2.1.4), some functional components include permitted operations which may require the PP or ST author to tailor the security requirement as appropriate for the PP or ST. These operations are:

a) *assignment*, allowing the specification of an identified parameter;

b) *iteration*, allowing multiple use of the same functional component to express different requirements;

c) *selection*, allowing the specification of one or more elements from a given list;

d) *refinement*, allowing the addition of details to the security requirement, thereby restricting the set of acceptable solutions without introducing any new dependencies on other SFRs.

12.3.3.2 Iteration

The *iteration* operation is often needed to express SFRs using components in the FMT (Security Management) class, which are called up as dependencies by many different functional components in ISO/IEC 15408-2. In order to satisfy such dependencies, it will typically be necessary to use the same component, with the assignment and selection operations completed differently. For example, FMT_MSA.1 may be iterated a number of times to define distinct SFRs relating to the management of different types of security attributes. Similarly, it may be desirable to make multiple use of components from the FDP_ACC and FDP_ACF families in the case where a TOE is required to enforce different access control policies, e.g. Discretionary Access Control (DAC) and Role Based Access Control (RBAC).

You are encouraged to use the iteration operation where the clarity of the PP or ST can be enhanced, e.g. to break down a complex and unwieldy SFR into distinct and manageable functional requirements. Use of the iteration operation does, however, pose other potential problems when presenting the SFRs in the PP or ST.

12.3.3.3 Assignment and selection

In an assignment there is the possibility that the value of the parameter may be null, whereas with a selection there is always at least one value of the parameter identified. By completing an assignment or selection operation in a PP removes any decision by the ST author as to how the functional component is to be tailored to meet the security objectives (other than the possibility of refinement). In other words, there are no aspects (insofar as the operation is concerned) that are 'to be defined' by the ST author.

Generally individual *assignments* or *selections* will require completion by the ST author. In a PP, over-qualification through completion of operations, or too much detail, may unduly restrict the number of TOEs that might be able to claim conformance with the PP. The balance of completing operations is based on the need for a PP to be:

a) a complete set of the requirements of the author;

b) implementation-independent;

c) sufficiently detailed to demonstrate that the objectives are met.

Therefore, it is necessary to complete assignment and selection operations to the extent needed to meet the security objectives. A critical test will come when you construct the security requirements rationale: the arguments you present to demonstrate the suitability of the IT security requirements to meet the security objectives should not rely on details that have not been specified in the SFRs. For example, in the case of an access control SFR based on FDP_ACF.1, you may consider it appropriate to leave the specification of access control rules entirely in the hands of the ST author if such rules are already defined in an OSP which the relevant (access control) security objective is intended to meet. In this case a PP author should complete the assignment and selection operations only as far as required to satisfy the general security objective, leaving sufficient freedom to the author of an ST that claims compliance to the PP to define the specifics of the access control rules implemented in the TOE.

One technique that you may use in order to solve the above problem is that of *partially* completing the operations. By adopting this approach you can give maximum flexibility to the ST author, whilst at the same time precluding potential choices for assignments or selections that would not be consistent with the security objectives for the TOE.

For example, in the following SFR (based on FAU_STG.4.1), the selection operation has been partially completed by precluding selection of the option 'ignore auditable events', which the PP author has judged to be inconsistent with the security objectives for the TOE. The SFR therefore presents the ST author with a choice of two (rather than three) acceptable options:

*The TSF shall [selection: **‘prevent auditable events, except those taken by the authorised user with special rights’, ‘overwrite the oldest stored audit records’**] and [assignment: **other actions to be taken in case of audit storage failure**] if the audit trail is full.*

With assignments, the PP author may wish to limit the choices an ST author can make to a set of options acceptable for the environment. In this case, the PP author may wish to complete the assignment operation by turning it into a selection operation containing the valid choices, which in turn can be completed by the ST author.

As a general principle, a *partially* completed selection is valid if the set of options it presents is a subset of the options that are permitted by the original functional component. Similarly, a *partially* completed assignment is valid if the permitted values to complete the assignment are also valid assignments with respect to the original functional component. If for any reason these conditions are not met, then you have ended up with an extended functional component with a different assignment or selection operation.

Completing the operations of assignment and selection is reasonably straightforward. In the case of assignment, you simply need to ensure that the parameter is specified unambiguously. In the case of selection, you simply need to select the appropriate item(s), based on consideration of the security objectives for the TOE. You should, however, consult the guidance given in the annexes to ISO/IEC 15408-2 if in doubt.

Where assignment or selection has been performed in a PP, it is mandatory that you highlight the text that has been specified (this is helpful to the reader, and especially to the PP evaluator checking conformance to ISO/IEC 15408). The customary way of highlighting is by using italics, but bolding or a different character set can also be used.

For example, FMT_SAE.1.1 could be presented as:

*The TSF shall restrict the capability to specify an expiration time for **a user's password to the authorised administrator**.*

In this case bold has been used for highlighting, since, being an example, the text is already in italics.

If an operation is left uncompleted, it is mandatory for the ST author to complete the operation.

Any uncompleted (or partially completed) operations should, where appropriate, be accompanied by an explanation, targeted at the ST author, of how the operation should be completed (for example, in the form of an application note). It may be helpful to make it clear that the onus is on the ST author to specify the details. For example, FDP_RIP.1.1 could be specified in a PP as:

*The TSF shall ensure that any previous information content of a resource is made unavailable upon the **allocation of the resource to** the following objects [assignment: **list of objects specified by the ST author**].*

For each SFR you have included in the PP, you need to make a judgement as to whether to complete any *assignments* or *selections* included in the functional component used to express the SFR. In a ST all assignments and selections need to be completed.

12.3.3.4 Refinement

For each SFR you have included in a PP or ST, you need to make a judgement as to whether to specify any *refinement* of the SFR.

The operation of refinement may be performed on any functional component element, and involves specifying additional technical details which do not levy any new requirements to those specified in the text, but rather restrict the set of acceptable implementations. A refinement is acceptable if meeting the refined requirement also means meeting the unrefined requirement. Use of refinement may be appropriate in the following circumstances:

a) where the PP is being written by an organisation which has additional technical details, such as organisation policy information, not included in the appropriate ISO/IEC 15408-2 component;

b) where the selected functional component would permit implementations which would not make sense, or would otherwise be inappropriate, for the type of TOE considered, unless it is refined so as to exclude that possibility e.g. on the grounds of interoperability;

c) where the readability of the SFR may be improved.

As with assignment and selection operations, it is recommended that you highlight the text that has been refined to assist the reader (and the PP evaluator in particular).

An example of the use of the refinement operation is as follows (based on FMT_MTD.3.1):

The TSF shall ensure that only secure values are accepted for TSF data. ***Refinement: the TSF shall ensure that the minimum password length enforced by the TOE is configured to a value of at least 6 characters.***

12.3.4 How should the audit requirements be specified?

If the PP or ST includes auditing requirements (i.e. based on FAU_GEN.1) then ISO/IEC 15408 requires that the minimum set of events which must be auditable, and the minimum information which must be recorded, is specified through the consideration of all other functional requirements included in the PP or ST.

This selection will depend on a number of factors, including:

a) any security policy requirements on security audit, as defined in an OSP;

b) the importance of auditing in achieving the security objectives;

c) the relevance of potential events, and their characteristics, to the security objectives;

d) cost/benefit analysis.

For example, if the TOE is intended to defend against the actions of malicious users or hackers, it is likely that events such as login or access control violations will need to be auditable where the PP or ST includes such SFRs. However, events relating to the use of administrative functions may not need to be auditable, depending on the extent to which an administrator is (or has to be) trusted, in which case the trustworthiness of the administrator would be stated as an assumption.

The question of cost/benefit analysis may rest on such issues as:

a) is the benefit of collecting the information worth the impact on performance?

b) if the information is collected, will the administrator have sufficient resources (e.g. tool support) to effectively analyse the data?

c) what are the likely costs of managing or archiving the data collected?

ISO/IEC 15408 identifies three pre-defined levels of auditing, namely *minimum*, *basic*, or *detailed* (see ISO/IEC 15408-2, 2.1.2.5): for each such level, ISO/IEC 15408-2 tells you which events should be auditable (as a minimum), together with the minimum information to be recorded, based on the functional components included in the PP or ST (see also ISO/IEC 15408-2, C.2). These three levels can be broadly characterised as follows:

a) The *minimum* level typically requires only some defined subset of operations or events associated with a given functional component to be auditable. This subset is generally defined to be the most interesting or significant type of event.

b) The *basic* level typically requires all operations or events associated with a given functional component to be auditable, e.g. successful and unsuccessful login attempts.

c) The *detailed* level generally differs from the *basic* level by requiring additional information of interest to be recorded. This level is only likely to be appropriate where the amount of audit data generated is anticipated to be small, or if the data will be subject to analysis by sophisticated audit analysis tools or intrusion detection facilities.

If none of these levels is appropriate, you should select the *not specified* level, and list all required auditable events explicitly in FAU_GEN.1.1. For example, you might use the *minimum* level for guidance, but choose to deviate from the *minimum* requirements in specific cases because a different subset of operations or events is more relevant to the security objectives, e.g. if FDP_ACF.1 is included in the PP or ST, you may consider that unsuccessful access attempts should be auditable rather than *successful* attempts (which is what ISO/IEC 15408-2 requires for the *minimum* level).

You will need to compile a list of auditable events by going through each functional component used in turn; in the case of the pre-defined levels of *minimum, basic* or *detailed,* these are explicitly identified in the *Audit* section included for each family of components. It is recommended that you construct a table, identifying the events and (where appropriate) the additional information to be recorded, which can be referenced by FAU_GEN.1.1 and FAU_GEN.1.2 as appropriate.

12.3.5 How should management requirements be specified?

ISO/IEC 15408-2 identifies, in the *Management* section included for each family of components, a list of management activities which should be considered for the component. This may suggest the need to include particular components from the FMT (Security Management) class. However, it is important to note that this section is intended to be *informative*. There is therefore no need to justify any decision not to include particular management components in the PP or ST (unless, of course, they are explicitly identified in the *Dependencies* section within ISO/IEC 15408-2).

Generally speaking, possible management activities are identified where a functional component refers to, or implies the existence of, configurable TSF data which may need to be managed and controlled. For example, the security objectives for the TOE might be undermined if the ability to modify such data was not restricted to administrators of the TOE. Therefore FMT components are often included in order to define *supporting* SFRs, in order to ensure that the security objectives for the TOE are met, and that the SFRs as a whole are mutually supportive.

Management activities can be derived from the functional model of the TOE. Typical management activities that need to be considered are:

- Registration or de-registration of users

- Creation of objects

- Modifications of security attributes of users, objects, sessions etc.

- Changes in the behaviour of security functions (including starting and stopping all or part of the TOE functions)

- Modification of audit parameter

- Change of TSF internal state variables that are relevant for security (e. g. changing to maintenance mode)

You should consult the guidance on the FMT class given in ISO/IEC 15408-2, Annex H when choosing functional components from this class.

12.3.6 How should SFRs taken from a PP be specified?

Where an ST claims compliance with one or more PPs, it is likely that the SFRs will be specified either completely or mostly by the PP. In such cases, the ST author must decide whether to specify the PP functional requirements in full (in order to ensure all the text is in one place), or whether to simply reference the PP and specify SFRs where these differ from the PP.

The latter approach will simplify the ST but requires the reader to look at both the PP and the ST to get a full picture. The reader of an ST is more likely to be interested in the IT security functions than in the SFRs. This includes the evaluator of the TOE (since the content of evaluation evidence - such as design, test documentation and guidance documents - is likely to be more easily related to the IT security functions in the TOE summary specification than to the SFRs). The main purpose of specifying SFRs in an ST is to be able to demonstrate traceability back to relevant PPs, and to the SFRs as defined in ISO/IEC 15408-2. There is indeed a case for relegating the statement of SFRs to an annex so as not to confuse the reader by having two specifications of security functionality in the ST.

It should, however, be noted that some SFRs in the PP may have operations (such as assignment or selection) that are left to the ST author. In such cases it is recommended that the SFR is specified in full, with the completed operations emphasised by suitable typesetting (e.g. using italics). Any necessary explanations should be added using the same typesetting. Such an approach will make it easier for the reader of the ST (and the ST evaluator in particular) to see which operations have been performed, and in which manner. It will also facilitate the construction of the ST rationale.

12.3.7 How should SFRs not in a PP be specified?

In some cases it will be necessary to specify SFRs in an ST where these are not in a corresponding PP. This may be necessary where:

a) there is no appropriate PP available for the TOE to claim compliance with;

b) the sponsor considers that the benefit to be gained by having functional or assurance requirements, that are in addition to what is required by the PP, is sufficient to justify the additional cost that would be incurred.

In such cases, the approach to the specification of SFRs is the same as described in the preceding clause. Where SFRs are specified in addition to those required by a PP, the ST author must ensure that these do not conflict with SFRs in the PP (the ST rationale will need to demonstrate that such conflict does not occur).

12.3.8 How should SFRs not included in Part 2 of ISO/IEC 15408 be specified?

ISO/IEC 15408 requires that if the PP or ST author wishes to include a functional requirement for which there is no appropriate functional component defined in ISO/IEC 15408-2, the resultant SFR should be specified using Part 2 components as a model for presentation.

The decision as to whether there is an appropriate functional component in ISO/IEC 15408-2 to use can be a difficult one to make, since this requires a high degree of familiarity with its content. It is recommended that you consult the guidance in subclause 12.3.2 which identifies the appropriate functional components from ISO/IEC 15408-2 to express common security functional requirements. It is often the case that the desired security functional requirement can be obtained through appropriate application of the refinement operation, or through permitted assignment or selection operations. However, it is recommended that you do not attempt to ‘shoehorn’ a security functional requirement into a functional component if this does not readily lead to the SFR you want, i.e. it results in an SFR whose meaning or intent cannot be readily discerned by the reader, or which (through the use of an inappropriate component) introduces inappropriate dependencies that need to be argued away.

Guidance how to define an extended component correctly is provided in clause 11.

12.3.9 How should the SFRs be presented?

Writing a set of SFRs that are demonstrably compliant with the requirements of ISO/IEC 15408 is not (of course) the *only* aim of the PP or ST author. You should also consider how best to present and express the SFRs such that the general reader can understand what the security requirements mean. There are a number of steps you can take to enhance readability, without compromising compliance with ISO/IEC 15408.

Firstly, group the SFRs under headings which are appropriate for your PP or ST: do not feel constrained to adopt the class, family or component headings used in ISO/IEC 15408-2.

Secondly, do not feel constrained to adopt the functional element labelling system used in ISO/IEC 15408-2 for labelling the SFRs in your PP or ST. It is perfectly acceptable to adopt your own labelling system (which may feature more meaningful labels), provided the mapping of SFRs onto the relevant functional component from ISO/IEC 15408-2 is demonstrated (e.g. in an annex). Indeed, such an approach is likely to be highly desirable where the PP or ST includes functional components which are invoked several times. This is because the alternative is to have SFRs that do not have unique labels: the lack of unique labels for SFRs presents significant problems when constructing the security requirements rationale.

Thirdly, judicious use of the refinement operation may improve the readability of the SFR by substituting generic terms (such as *security attributes*) with more specific terminology relevant to the type of TOE or security functionality being described. For example, the following SFR is based on FMT_MSA.3.1:

*The TSF shall enforce the **DAC policy** to provide **restrictive** default values for **object permissions**.*

In this example, refinement has been used to replace the generic ‘security attributes that are used to enforce the SFP’ with the policy-specific ‘object permissions’.

Any such use of the refinement operation should be clearly highlighted and explained in the PP or ST Rationale (to support evaluation of the PP or ST).

12.3.10 How to develop the security requirements rationale

Unless the Security Target or Protection Profile is a low assurance one (which is explained in more detail in clause 15.1), a rationale is required that explains how the security objectives are met by the security functional requirements. This rationale needs to trace all security objectives to the security functional requirements that together shall address the objective. This tracing needs to show that each security functional requirement traces back to at least one security objective and each security objective has at least one security functional requirement tracing to it.

In most cases a single security objective will trace to more than one security functional requirement and often a single security functional requirement will support more than one security objective. In most Security Targets and Protection Profiles the number of security functional requirements will be larger than the number of security objectives, since the security objectives will be more generic than the security functional requirements. For example a security objective:

The TOE will ensure that each user is uniquely identified, and that the claimed identity is authenticated, before the user is granted access to the TOE facilities.

will be most likely mapped to a number of security functional requirements, specifying:

- how users are identified

- how users are authenticated

- what happens if the authentication fails

- how users and their authentication data are created and managed

- how users are bound to subjects

More important than just the tracing of security objectives to security functional requirements is the justification that the sum of security functional requirements a security objective is traced to completely satisfies the objective. In the example stated above this justification may be easy to derive, but this is not true for all security objectives. Especially in the case of security objectives that specify properties of the TOE, the justification that the security functional requirements completely address the security objective may be non-trivial. For example with a security objective:

The TOE shall ensure that no information can flow from a subject operating with a specific security label to a subject operating with a hierarchically lower security label or an incompatible security label

It is hard to justify that the security functional requirements for a lattice based mandatory access control policy **completely** address the security objective. Additional security functional requirements may be added e. g. for an architecture that provides better support for information flow control but still it will not be possible to show that the sum of the security functional requirements addresses the security objective completely. In the example even with all the security functional requirements being correctly implemented, covert channels may still exist that allow information to flow in a way that contradicts the security objective. The justification for completeness provided as part of the security requirements rationale should acknowledge this and indicate that within the TOE model provided by the sum of the security functional requirements the security objective is completely addressed, i. e. provide justification that there is no security functional requirement that contradicts the security objective.

In general one can say that the tracing between security objectives and security functional requirements and the justification for completeness becomes easier when the security objectives are expressed as functions rather than properties and to a level of detail that is close to the one provided in the security functional requirements. Security objectives should therefore be already as precise as possible. When writing a Security Target or Protection Profile it is wise to reconsider the security objectives and attempt to formulate them more precisely in the case one has problems with tracing the security objectives to security functional requirements or with the justification that the security functional requirements completely address the security objectives.

12.4 How to specify assurance requirements in a PP or ST

12.4.1 How should security assurance requirements be selected?

The selection of the assurance requirements will require the balancing of several factors including:

a) the value of the assets to be protected and the perceived risk of compromise of those assets;

b) technical feasibility;

c) likely development and evaluation costs;

d) required timescales for development and evaluation of the TOE;

e) perceived market requirement (in the case of products);

f) any identified dependencies of functional components on assurance components.

The greater the value of the assets to be protected, and the greater the risk to those assets, the higher the level of assurance that will be required in the security functions used to protect those assets. This should be reflected in the statement of security objectives. Organisations may define their own policies and rules to determine the level of assurance that is needed to ensure that the risks to their assets are reduced to an acceptable level. This may in turn define the required level of assurance in products to be used within that organisation.

Other factors such as costs and timescales will tend to act as a constraint on the level of assurance that is actually achievable in practice. Technical feasibility will be a factor where it is considered impractical to generate the evidence required by specific assurance components. This may be highly relevant for legacy systems (where design documentation is unavailable), or where a high assurance level is ideally required, but it is not technically feasible to generate the required semi-formal or formal evidence within acceptable timescales. Wherever there are practical constraints on the assurance that may be achieved, it may be necessary to accept that the maximum assurance attainable is less than the ideal. Such acceptance of risk should, again, be reflected in the statement of security objectives.

The statement of security objectives may also indicate a need for specific assurance requirements which should be included in the SARs. For example:

a) The security objectives for the TOE may state that the TOE should be resistant to attackers who have a high attack potential. This would be a clear pointer to the inclusion of AVA_VAN.5 which requires such resistance to be demonstrated.

b) The security objectives may indicate that self-protection, domain separation, or non-bypassability are a concern, in which case it is necessary to include the ADV_ARC.1 component. Note that the class ADV_ARC has only one component, but the level of architectural description required depends on the component chosen from the ADV_TDS class.

c) The security objectives may note that the security of the TOE is critically dependent on the security of the development environment. This would strongly suggest that the SARs should include a component from the ALC_DVS family to ensure that the security of the development environment is examined.

The selection of the SARs will relatively straightforward where it involves simply choosing an appropriate assurance package, such as an ISO/IEC 15408 EAL. The definitions and descriptions of the assurance package should be consulted to ensure that the package is appropriate given the statement of security objectives (e.g. in the case of the EALs, see ISO/IEC 15408-3, Clause 6). It is possible that an assurance package exists that provides broadly the level of assurance that is needed, but is lacking in specific areas when measured against the security objectives. In such cases it would be appropriate to include augmented assurance requirements (i.e. requirements that are additional to those mandated by the package) in order to ensure that the security objectives are satisfied.

Where augmented assurance requirements are specified, the PP or ST author should ensure that the assurance component dependencies are satisfied for the additional requirements. For example, if a PP or ST augments EAL3 with AVA_VAN.3, then it should also augment with ADV_TDS.3 and ADV_IMP.1, as these are not included in EAL3. In addition ADV_FSP.4 needs to be included, since ADV_TDS.3 has a dependency on ADV_FSP.4.

12.4.2 How to perform operations on security assurance requirements

The following operations are possible:

a) *iteration*, allowing multiple use of the same assurance component;

b) *refinement*, allowing the addition of details to the assurance requirement without introducing any new dependencies on other SARs;

c) *assignment*, allowing to assign values to an element of a SAR with an assignment parameter.

In practice, the iteration operation would only be used where it is necessary to apply different refinements to the same assurance component which apply to different parts of a TOE, or where a PP or ST specifies different sets of assurance requirements for different parts of a composed TOE (see 14.1). In the latter case, iteration would be necessary for assurance components (whether refined or not) that apply to more than one part of the composed TOE.

The refinement operation on SARs might be used to:

a) constrain the developer actions by mandating such things as the use of specific development tools, methodologies, life-cycle models, analysis techniques, notations, adherence to specific standards, and so on;

b) constrain the performance of the evaluator actions, e.g.:

   - in the case of ADV_IMP.1, specifying which parts of the TOE implementation representation should be included in the subset examined

   - in the case of AVA_VAN.1, identifying specific a minimum set of public domain sources of vulnerabilities that need to be considered since they usually describe vulnerabilities that are relevant in the context of the TOE.

There are only two instances in the list of SARs in ISO/IEC 15408-3 where assignments are allowed: ADV_INT.1.1D and ADV_SPM.1.1D. In the first case the PP/ST author needs to define with the assignment the subset of the TSF for which the element applies, in the second case the PP/ST author needs to define with the assignment the set of security policies that are formally modelled.

12.4.3 How should SARs not included in Part 3 of ISO/IEC 15408 be specified in a PP or ST?

ISO/IEC 15408 requires that if the PP or ST author wishes to include an extended SAR for which there is no appropriate assurance component defined in ISO/IEC 15408-3, the resultant extended SAR should be specified using ISO/IEC 15408-3 components as a model for presentation. Clause 11 provides guidance how to define such an extended assurance component in a ST or PP.

12.4.4 Security assurance requirements rationale

The structure of a Protection Profile and Security Target (unless it is a low assurance one as explained in clause 15.1) also requires a rationale why the set of security assurance requirements has been chosen. As figure 3 shows the security assurance requirements are not required to be derived from the security problem definition or the security objectives and therefore may be derived from other sources. ISO/IEC 15408-1 therefore allows providing no explanation how the security assurance requirements have been derived or point to specific regulations that require a specific set of security assurance requirements.

In many cases the security assurance requirements have been derived based on the threats and the threat agents identified in the security problem definition with the intention to select the security assurance requirements such that the TOE can be expected to be resistant against attacks launched by threat agents included in the security problem definition. If this is the case, one should express this in the rationale for the selection of security assurance requirements.

13 The TOE summary specification

A TOE summary specification is required for a Security Target, but not for a Protection Profile. This clause therefore applies only to Security Targets.

The purpose of the TOE summary specification is to provide consumers with description of the security functions of the TOE that explains how the SFRs are met. The TOE summary specification should therefore describe the security functions in the context of the overall functionality and architecture of the TOE and provide sufficient detail to get an abstract view of the overall TOE and how the TOE implements the SFRs.

The TOE summary specification therefore presents a security centric abstract model of the overall TOE where the subjects, objects, security attributes and rules as defined in the SFRs are put into the context of the TOE architecture and its general functionality. This model may still abstract from quite a number of non-security functions the TOE provides, as long as those functions are unrelated to the security functionality implemented by the TOE. The level of detail presented in the TOE summary specification should be higher than the level of detail in the TOE description, and focus on the explanation how the SFRs are met. A mapping is required that shows how the SFRs are met by the security functionality described in the TOE summary specification.

A good way to structure the TOE summary specification is to start with a general overview, which presents an abstraction of the TOE architecture including the TSF boundary. It is beneficial to describe also how the TSF protects itself from tampering and bypass, even if conformance to ASE_TSS.2 is not required. Then the security functions should be described based on the functional model that was used to derive the SFRs. It is a good practice to write the TOE summary specification and the chapter describing the SFRs in parallel ensuring that each SFR is derived consistent with the functional model and thereby constructing the mapping from the security functions described in the TOE summary specification to the SFRs. The TOE summary specification should basically contain the functional model (as derived using the guidance in section 12.2.) enhanced by text that puts this model into the context of the whole TOE with all its functions and its architecture. This gives the reader also an understanding why specific security functions or details of security functions have selected and how they support the overall functionality of the TOE. In addition the mapping of the TSS to the SFRs is automatically provided since both are derived from the same model.

In the case of a composed TOE, the TOE summary specification needs to describe the individual components and how they interact to provide the SFRs. The description shall provide the reader with an understanding how the SFRs for the composed TOE can be mapped to the SFRs of the components and how those SFRs interoperate. For more guidance on Security Targets for composed TOEs see clause 14.1.

14 Specifying PP/STs for composed and component TOEs

14.1 Composed TOEs

Within their operational environment most of the TOEs will interact with other IT products or systems. In many cases they will need the support of such IT products or system to satisfy the security functional requirements. A simple example is a database system TOE that relies on the file protection, address space separation and user authentication functions of the underlying operating system. Another example is an operating system that relies on an external LDAP server for storing digital certificates and certificate revocation lists used for authentication; it also relies on an external PKI to generate those certificates and certificate revocation lists and publish those in a timely manner via the LDAP server. When combining the two examples, the database management system (through its dependency on the user authentication on the operating system) also relies on the LDAP server and the PKI system for user authentication. This example again can be easily extended when also a smartcard is used within the user authentication process. In this case dependencies exist on the smartcard itself but also on the system used for smartcard personalization.

Those examples show that an at a first glance simple security functional requirement (user authentication) may require the correct and secure cooperation of a number of IT products that themselves may well be evaluated separately. This section deals with the problem of specifying security functional requirements for the TOE in conjunction with security objectives for the TOE environment in order to address this problem of security functional requirements being satisfied by a combination of IT products.

In the examples stated above we have the following dependencies:

- The database management system relies on the operating system for user authentication, file protection and address space separation

- The operating system relies on the underlying hardware for address space separation and protection against direct access of unprivileged programs to attached I/O devices and dedicated processor configuration register

- The operating system relies on the LDAP server for protecting the information used for user authentication against unauthorized access. It also relies on the LDAP server for providing the information upon request in a timely way that also protects the information from undetected modification when transferred between the LDAP server and the operating system.

- The operating system relies on the PKI for generating digital certificates with correct information about the user bound to the certificate and for managing those certificates correctly (including timely publishing certificates as well as certificate revocation lists on the LDAP server).

- The operating system relies on the smartcard for protecting the user's private key and for using this key only after it has received the correct authentication information (in our example the PIN).

- The smartcard relies on the operating system for protecting the user's PIN from the time the user has entered it, when transferred to the smartcard, and up to the time the PIN is securely erased from the memory of the operating system. The smartcard relies on the operating system for not misusing the user's PIN e. g. submitting it to the smartcard without being authorized by the user to do so.

This is just a list of dependencies we will use to demonstrate how they can be handled in a PP or ST.

When analyzing the dependencies one can easily identify:

- The database depends on the operating system to provide its security functionality

- The operating system depends on the hardware

- The operating system depends on the LDAP server

- The operating system depends on the PKI

- The operating system depends on the smartcard

- The smartcard depends on the operating system

In the case where one component depends on another component, ISO/IEC 15408 calls them the "dependent" and the "base" component. In our example in the combination database and operating system, the database is the dependent component and the operating system is the base component. Similar in the combination operating system and hardware the operating system is the dependent component and the hardware is the base component. In the case of the smartcard and the operating system, both components depend on each other and are therefore both depend and base components.

When developing a Security Target or Protection Profile for a dependent component, the dependencies on the base component have to be addressed as assumptions and the security objectives for the operational environment derived from those assumptions. In the example of the DBMS the assumptions could be:

- *Assumption_1*
  The operational environment will protect the DBMS software from interference and tampering by any other application software executing on the same system as the DBMS

- *Assumption_2*
  The operational environment will protect the files used by the DBMS to store its user and TSF data from unauthorized access

- *Assumption_3*
  The operational environment will identify and authenticate individual users and provide a way for the DBMS to obtain the identity of the user on behalf of which a request to the DBMS is made

Those assumptions can now be used to specify quite precise objectives for the operating system as part of the operational environment. The level of detail in those objectives depends very much on the specific requirements on the DBMS. For example if audit is a security functional requirement for the DBMS it may be useful to require also some level of auditing from the operating system in order to detect attempts to bypass or tamper with the security functionality of the operating system the DBMS depends on. An example of security objectives derived from the assumptions above is:

- *The operating system shall provide a mechanism that allows the DBMS to be executed in its own execution domain that is protected from interference and tampering by other application programs executing under the control of the operating system*

- *The operating system shall protect the executable programs that belong to the DBMS from unauthorized access*

- *The operating system shall provide a mechanism that allows to detect integrity violations of the DBMS software and prohibit starting the DBMS when an integrity violation is detected that can not be corrected*

- *The operating system shall provide an access control mechanism for files that differentiates at least between read and write/update access and allows to individually define the level of access (including "no access") down to the granularity of an individual user*

- *The operating system shall allow restricting the management of the access rights for files to individual users or defined groups of users*

- *The operating system shall identify and authenticate individual users before they can call DBMS functions*

- *The operating system shall use a protected mechanism for authentication that limits the probability of falsely authenticating a user to a probability of less than 1/1,000,000*

- *The operating system shall have the ability to audit successful and unsuccessful authentication attempts where the audit record contains the identity of the user and the time and date of the authentication attempt*

- *The operating system shall provide an interface that the DBMS can use to correctly obtain the identity of the user on behalf of which a database function is called*

One can see that most security objectives can be easily mapped to security functional requirements as defined in ISO/IEC 15408-2. Only the first security objective is different since it addresses the architectural property of domain separation. The Security Architecture documentation needs to describe how this property is implemented by the operating system. Security Architecture documentation is mandatory for assurance levels of EAL2 and higher.

In a case like the one we have with the DBMS as a dependent component and the operating system as the base component, one can define the security objectives for the operating system with a quite high level of detail very close to the level of detail in security functional requirements. Whenever this is possible one should provide such a level of detail.

There are other situations where the assumptions and also the security objectives for the operational environment derived from those assumptions have to be more generic. If we take the example of the operating system as the dependent component and the LDAP server as the base component, the assumptions we need to make are:

- *The operational environment shall protect the digital certificates and certificate revocation lists required by the operating system for user authentication from unauthorized modification and from unauthorized addition of certificates and certificate revocation lists*

In this case the ST or PP may well leave some of the details of this protection open to allow for different ways to satisfy this assumption. Security objectives for the operational environment derived from this assumption may be:

- *The LDAP server shall identify and authenticate users before allowing modification and/or addition of certificates and CRLs used for user authentication by the operating system*

- *The operational environment shall protect data communicated between the LDAP server and the operating system from undetected modification (including addition and replay)*

In this example one probably does not want to specify the way how those security objectives for the operational environment are achieved in more detail allowing for a range of different ways to comply with those requirements. In the example given it is deliberately left open if the second security objective is achieved using a cryptographically protected communication protocol or if this objective is achieved by a physically protected network.

When developing a Security Target or Protection Profile for a dependent component, one has to differentiate between the case where the base component has been evaluated and the evaluation results are available when evaluating the dependent component and the case where either the base component is not evaluated or where the evaluation results of the base component are not available.

For the first case ISO/IEC 15408-3 includes the ACO assurance class, which defines the evaluation criteria for the composition of evaluated components. The author of a Security Target or Protection Profile for a composed TOE has to include the components from ACO class that he feels are appropriate for the level of assurance. To assist this ISO/IEC 15408-3 has defined three "component assurance packages" that may be included in a Security Target or Protection Profile for a composed TOE. If one decides to select the components from the ACO class different from those predefined packages, one has to ensure that the dependencies are resolved.

14.2 Component TOEs

While there are TOEs that are self-sufficient with no explicit reliance on another IT component in their environment, there are also a number of TOEs where this is not the case. Those are specified as "component TOEs" in a Security Target or Protection Profile. Typical examples are:

- a software package that provides defined security functions but is intended to be integrated into a number of different products. The software package relies on the product it is integrated into for the protection of its TSF and TSF data as well as the management of some TSF data.

- an application that implements access control on its own objects but relies on user identification and authentication to be provided by the environment.

- an application or an operating system that relies on a cryptographic coprocessor for the provision of cryptographic functions and the management of their keys.

In all those cases one or more security objectives are only partly mapped to security functional requirements of the TOE and partly mapped to the environment. The TOE therefore can only be evaluated under the assumption that the environment correctly implements security functional requirements that completely address the part of the security objectives that the TOE is not able to address itself.

A Security Target or a Protection Profile for a component is therefore not much different from the one for a self-contained product. The only difference is that the security objectives for the IT environment that are required to be satisfied to completely address the TOE security objectives are usually specified quite precisely mentioning also (where possible) the type of IT product in the environment that is intended to satisfy the objective. In the example given in clause 14.1 the Security Target for the operating system can clearly and separately define the security objectives to be satisfied by the underlying hardware, the LDAP server, the PKI system and the smartcard. Those security objectives should be defined as precise as possible such that they can be easily mapped to security functional requirements as specified in the Security Targets of those components. This allows an easier mapping when evaluating the sum of those components as a composed TOE.

15 Special cases

15.1 Low assurance Protection Profiles and Security Targets

In the case of a Protection Profile or Security Target where the assurance requirements are not higher than those defined for the EAL 1 assurance level, ISO 15408 allows to simplify the Protection Profile or Security Target. In the case of a Security Target or a Protection Profile one may omit:

- The security problem definition
- The security objectives
- The security objectives rationale
- The security requirements rationale except for the justification of unresolved dependencies between security requirements as defined in ISO 15408.

This allows for simpler Security Targets and Protection Profiles that target low assurance products. All other sections need to be addressed as described before.

A low assurance Security Target or Protection Profile may only claim conformance with a low assurance Protection Profile, but a non low-assurance Protection Profile or Security Target may claim conformance to a low-assurance Protection Profile. Such a non low-assurance Protection Profile or Security Target needs to include all sections mandatory for a non low-assurance Protection Profile or Security Target. Therefore the sections omitted in the low-assurance Protection Profile they claim conformance with must be included in the non low-assurance Protection Profile or Security Target.

15.2 Conforming to national interpretations

In addition to the requirements defined in ISO/IEC 15408 for Protection Profiles and Security Targets, individual national schemes may define specific national interpretations of ISO 15408 that need to be addressed in order to get a Protection Profile or Security Target evaluated under the national scheme. Usually those national interpretations are published by the national scheme and the author of a Protection Profile or Security Target that intends to use a specific national scheme should contact the scheme for such interpretations.

In addition to national interpretations there may be joint interpretations including modifications to the text of ISO/IEC 15408 that are accepted by the group of nations that cooperate to achieve mutual recognition of certifications. Such interpretations also need to be considered when developing a Protection Profile or Security Target and the author is advised to contact a scheme participating in this cooperation to obtain those joint interpretations related to the structure and content of a Security Target or Protection Profile.

15.3 Functional and assurance packages

In addition to Protection Profiles, ISO/IEC 15408 also allows the definition of functional and assurance packages. A functional package contains a set of security functional requirements and an assurance package contains a set of security assurance requirements. Mixed packages that contain both security functional and security assurance requirements are not allowed.

Such a package should have a name that allows identifying the package and it should contain a set of requirements that are useful and effective. For example a functional package may contain security functional requirements addressing one specific security aspect. A typical example is a functional package defining the functionality related to audit (minimum set of events that can be audited, protection of the audit trail, audit review requirements, audit management) without addressing any other aspect. Such a functional package may then be reused for different types of security products (e. g. operating systems, database management systems, firewall systems). When defining such a functional or assurance package, it is wise to address the dependencies either directly within the package or by providing advice how unresolved dependencies should be addressed when using the package.

16 Use of automated tools

The structured nature of ISO/IEC 15408, and the fixed content of an PP/ST, have raised the question as to what extent automated tools can help in the production and evaluation of key ISO/IEC 15408 documents such as PPs and STs.

The motivations for such tools range from allowing the developer to concentrate on the content of the PP/ST, taking care for him of the presentation burdens of an PP/ST, and of the resource consuming tasks, such as relationships/rational closures, or functional dependencies checks, to alleviating the evaluators from the most time consuming, less rewarding activities in the PP/ST evaluation.

Some attempts have released to the public usage tools to develop PPs/STs, which allow to derive an PP/ST from the security problem analysis down to the security requirements like "The CCToolBox" [9], or the other way around, from the security characteristics of a TOE to the security problem definition, like "The PPhelper" [10], both supported by a database of interrelated security concepts.

These attempts were discontinued, and have had not too much success.

Some commercial tools are available that draw on the principles of these initial publicly available proofs of concept. These are still available, and offer different functionality and checks on a formulated PP/ST.

The new formulation of ISO/IEC 15408 and the Common Criteria in XML form allow a new range of possibilities in the automation of PP/ST development, maintenance and integration into a developer's development life cycle and tools. These new possibilities do not have to rely on external or commercial tools, but are at the reach of the proper TOE developer.

How far can or should this tool support go in the production of evaluation evidence, and in particular, to the production of PPs and STs?

The clear guidance here is first to acquire the knowledge and concepts required to write and master these PPs/STs. Then, automation of some of the content and presentation requirements, easy copy of text extracted from the latest version of ISO/IEC 15408, and so on, can be made easy with some tool support. Expecting that the reverse approach, i.e., adopting a tool under the hope that it will provide useful PPs/STs, will mostly lead to disappointment.

Annex A
(informative)

Example for the definition of an extended component

The following section provides an example of the definition of an extended security functional component that attempts to address the requirement for recovery of TSF data. This example shows the way how to define and structure the definition of an extended component as well as the rationale for the component as it would be expected to be defined in a ST or PP.

TSF Data Recovery (FPT_REC_EXT)

Family Behaviour

TSF data recovery allows defining checkpoints for defined TSF data and later a recovery of this TSF data as it was at the time the checkpoint was set. This allows recovering TSF data after it has been modified e. g. by an administrator error or after it has been corrupted by some hardware or software malfunction.

Component levelling

FPT_REC_EXT.1 Basic TSF data recovery addresses a need to set checkpoints and recover from checkpoints by explicit administrator action.

FPT_REC_EXT.2 Advanced TSF data recovery addresses the need to set checkpoints and recover from checkpoints either automatically or by explicit administrator action.

Management: FPT_REC_EXT.1, FPT_REC_EXT.2

The following actions could be considered for the management functions in FMT:

- The management of the privileges required to define checkpoints and/or initiate recovery.

- The management of the TSF data that is included in the data saved at a checkpoint.

Audit: FPT_REC_EXT.1, FPT_REC_EXT.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

- Minimal: All successful recovery operations.

- Basic: All attempts to perform a recovery operation.

- Detailed: All attempts to perform a recovery operation, all checkpoint operations.

FPT_REC_EXT.1 Basic TSF data recovery

Hierarchical to: No other components.

Dependencies: FMT_SMR.1 Security Roles, FMT_MOF.1 Management of security functions behaviour

FPT_REC_EXT.1.1 The TSF shall allow user with the roles [assignment: list of roles] to define checkpoints for [assignment: list of TSF data]. The TSF shall store this TSF data when such a checkpoint is reached to a location where it can be recovered later.

FPT_REC_EXT.1.2 The TSF shall allow user with the roles [assignment: list of roles] to recover the TSF data from the values it had at the checkpoint.

FPT_REC_EXT.1.3 The TSF shall perform the following actions to ensure the consistency and integrity of TSF data after recovery [assignment: list of actions for consistency and integrity checks]

FPT_REC_EXT.1.4 The TSF shall perform the following actions when it detects that the TSF data is inconsistent or violates integrity during recovery [assignment: list of actions]

FPT_REC_EXT.2 Automated TSF data recovery

Hierarchical to: FPT_REC_EXT.1.

Dependencies: FMT_SMR.1 Security Roles, FMT_MOF.1 Management of security functions behaviour

FPT_REC_EXT.2.1 The TSF shall define checkpoints for [assignment: list of TSF data] under the following conditions [selection: at administrator defined time intervals, when the following conditions are satisfied [assignment: list of conditions], [assignment: other criteria]]. The TSF shall store this TSF data when such a checkpoint is reached to a location where it can be recovered later.

FPT_REC_EXT.2.2 The TSF shall recover the TSF data from the values it had at the checkpoint under the following conditions [assignment: list of conditions].

FPT_REC_EXT.2.3 The TSF shall perform the following actions to ensure the consistency and integrity of TSF data after recovery [assignment: list of actions for consistency and integrity checks].

FPT_REC_EXT.2.4 The TSF shall perform the following actions when it detects that the TSF data is inconsistent or violates integrity during recovery [assignment: list of actions].

FPT_REC_EXT.2.5 The TSF shall use recovered TSF data only when its consistency and integrity is ensured.

Rationale for the definition of the extended component

No SFR has been defined in ISO/IEC 15408-2 for the recovery of TSF data stored at defined checkpoints. This kind of recovery is important to ensure the secure operation of our TOE. The functional requirement specifies the conditions under which the values of defined TSF data is "saved" as part of a "checkpoint" operation and later those values are restored under defined conditions from their saved values. It may well be the case that restoring just a defined set of TSF data may result in an inconsistent (and potentially insecure) state of the TSF. Also the integrity of stored TSF data has to be ensured. Therefore it may well be required to perform checks to ensure that the overall state of the TSF after recovery is consistent and secure and the recovered TSF data was not modified. The SFR therefore allows the specification of consistency and integrity checks that need to be performed after recovery and before the normal operation of the TOE continues.

If those consistency checks fail, the TOE may decide to take automatic corrective action, enter a maintenance mode that allows for manual correction by administrative users, or perform other actions in order to prohibit that the TOE gets into an insecure state. FPT_REC_EXT.2 does not allow using recovered TSF unless its consistency and integrity is assured.

This security functional requirement component is an extended component for the FPT class of security functional components as defined in ISO/IEC 15408-2.

Bibliography

[1] *Common Criteria for Information Technology Security Evaluation (3 parts),* Version 3.1, Common Criteria Management Board, 2007. Downloadable from http://www.commoncriteriaportal.org/ (link valid May 2008).

[2] *List of PPs*, Downloadable from http://www.commoncriteriaportal.org/ (link valid May 2008).

[3] *ISO/IEC 15292, Information technology – Security techniques – Protection Profile registration procedures*.

[4] *Software Reliability Measurement, Prediction and Application*, J. Musa, A. Ianino and K. Okumotu, MacGraw-Hill, 1987, ISBN: 9780070441194.

[5] *Software Risk Management: Principles and Practices*, B. W. Boehm, IEEE Software, January 1991.

[6] *A Security Engineering Process*, J. D. Weiss, Proceedings of the 14th National Computer Security Conference, Washington D.C., USA, October 1991.

[7] *Secrets and Lies*, B. Schneier, John Wiley & Sons, 2000, ISBN: 9780471253112.

[8] *The Security Development Lifecycle*, M. Howard and S. Lipner, Microsoft Press, 2006, ISBN: 9780735622142.

[9] The CC Toolbox. Downloadable from http://cctoolbox.sparta.com/ (link valid May 2008).

[10] The PP Helper. Presented at the Third International Common Criteria Conference, Ottawa, Canada, 13-14 May 2002.

Index