Bundesamt
für Sicherheit in der
Informationstechnik

The PP/ST Guide

August 2010

Version 2
Revision 0

Table of Contents

Table of Contents

1 Introduction

1 This introduction addresses the following key questions: What is this guide? Who are its target audiences? Why was it written? What is the structure of this guide?

1.1 What is this guide?

2 This is a guide to Protection Profiles (PPs) and Security Targets (STs): Two constructs that are used in the Common Criteria for Information Technology Security Evaluation (referred to as "Common Criteria" or "CC" in the rest of this guide) also known as ISO/IEC 15408.

3 This guide is not intended as an introduction to the CC itself. Readers who are interested in this are referred to the CC, in particular Part 1. The most relevant chapters in the CC are:

- CC Part 1, chapter 7 and annex A discuss aspects regarding STs;
- CC Part 1, chapter 9 and annexes B and D discuss aspects regarding PPs;
- CC Part 1, chapter 8 and annex C discuss aspects of functional requirements, which are relevant for both STs and PPs;
- The chapters concerning classes APE and ASE in CC Part 3 define requirements for PP evaluation and ST evaluation respectively.

4 The CC exists in several versions: This guide is targeted at CC v3.1. References to specific text or sections in the CC are based on CC 3.1, Revision 3.

5 This guide was developed on behalf of the German Federal Office for Information Security (BSI). The BSI therefore holds the copyright on this document.

1.2 Who are the target audiences of this guide?

6 This guide is intended for three different audiences:

- **PP/ST readers**: PPs and STs contain a lot of sections, filled with abbreviations and specific terminology. This guide should therefore assist readers in pinpointing the sections of interest for them and what they should find in those sections.
- **PP/ST authors**: These form the primary audience of this guide. This guide should assist them in writing PPs and STs that meet the Common Criteria, yet are (relatively) easy to read and understand.
- **PP/ST evaluators and certifiers**: While this guide is not absolutely necessary for these audiences, it should assist them in gaining a better understanding of PPs and STs.

1.3 Why was this guide written?

7 The German BSI recognized that Protection Profiles and Security Targets in CC v2.x had a number of problems associated with them:

- PPs/STs are somewhat abstract documents;
- PPs/STs are documents that are unique to the CC and produced specifically for the CC. PP/ST authors could not draw upon other documents or experiences;
- at the time of issuing CC v2, there were no example PPs/STs that everyone agreed upon;
- the CC v2 criteria and methodology are not always unambiguous and are not always sufficiently explained.

8 This has led to substantial divergence in the creation of PPs and STs in CC v2, and to the fact that PPs and STs in CC v2 could not always be consistently interpreted.

9 The goal of this guide is therefore to provide more insight in Protection Profiles and Security Targets, together with some definitive examples. Together with the improved criteria for PPs and STs in v3.1 this guide is intended to contribute to "better" (comparable, readable, more distinctive, canonical etc.) PPs and STs.

1.4 What is the structure of this guide?

10 This guide provides a structured approach to creating a PP or ST and examples to give the reader insight in what is meant with the sometimes abstract wording in the Common Criteria.

11 Readers with little experience regarding Common Criteria Protection Profiles or Security Targets should confine themselves to chapters 2 and 3 as these were specifically written for readers that have little to no experience with the Common Criteria. For all other chapters a reasonable amount of experience with the CC is a must.

12 *Chapter 2: What are Protection Profiles and Security Targets?* describes how PPs and STs are intended to be used in the purchase of products. This chapter aims itself at relative novices in the CC and therefore omits a significant amount of detail.

13 *Chapter 3: Reading Protection Profiles and Security Targets* provides instruction on how to read and understand Protection Profiles and Security Targets. This chapter is also aimed at relative novices and is not intended as a complete description of the content of PPs and STs.

14 *Chapter 4: Writing a Protection Profile* discusses the production of a Protection Profile in detail. It describes two methods: The "backward" method, aimed at quick production of PPs and the "explanation" method, aimed at producing higher quality PPs, albeit at a greater effort. The latter method is described at considerable length,

and uses a detailed running example (a firewall) to explain all the methods and rules of thumb and show their application.

15 *Chapter 5: Writing a Security Target* discusses the production of a Security Target in detail. As Protection Profiles and Security Targets have much in common, this chapter focuses on the differences between the two, and how the "explanation" method can be used to produce STs[1] as well.

[1] The CC, in particular in Part 1, and the CEM describe in detail, how an ST (or a PP) can achieve conformance to a PP. Therefore this aspect is not discussed in detail in this guide. Instead we concentrate on guidance for the creation of new text, which is written by the author of a PP or ST.

2 What are Protection Profiles and Security Targets?

2.1 Introduction

16 The main use of the CC is to assess the security of IT products. As there are many different types of IT products, and IT products are used in many different ways, and in many different environments, the notion of security is usually different for different IT products. The end result of a CC evaluation is therefore never "this IT product is secure", but is always "this product meets this security specification".

17 The CC has standardized 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.

18 The CC recognizes two different security specifications: Protection Profiles and Security Targets. The difference between these two is best explained by the role they are intended to play in a product purchasing process: That is a customer seeks to buy a product from a developer.

19 The notions of customer, developer and product are deliberately kept abstract. A customer is someone who wants to buy a product. A customer can be a single individual, an organization, a group of organizations, a government department etc.

20 A developer is someone who wants to sell a product. A developer can be a single programmer, a small company, a large company, a group of companies working together etc.

21 Finally, a product can range from a small application or a smart card to a large operating system or a computer system containing hundreds of components.

2.2 Product purchasing processes

22 When a 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 chapter 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 customers' needs. In the remainder of this chapter we will call this a *selection-based purchasing process*.

23 When IT security is important, these purchasing processes have an added difficulty in that 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 for him;
- And even harder to determine that if a product claims to have security properties, that this claim is true.

24 To assist a customer through a purchasing process and address the difficulties listed above, an evaluation of the product may be useful, and in this case, Protection Profiles and Security Targets play an important role.

25 In the next two sections, we will show how an evaluation may assist each type of process:

- Specification-based and
- Selection-based.

2.3 Specification-based purchasing processes

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

- The customer must determine his informal security requirements.
- The customer must transform these informal security requirements into a specification suitable as input for a developer.
- The developer must build a product based on this specification.

27 As the customer in the end wants to know that "this product is useful for me", the quality of each of these steps is important. Each step is discussed in the following sections.

2.3.1 Informal security requirements

28 The process of determining informal security requirements, that is the process of determining, which assets the customer wants to protect against which attackers, and which level of protection they need, is outside the scope of the Common Criteria and therefore outside the scope of this guide. However, this does not mean that this is unimportant or easy by any means.

29 Nevertheless, the Common Criteria assumes that the customer is somehow capable of defining his or her informal security requirements. If this is done incorrectly, the product that is purchased in the end may not be in the customers' interest.

2.3.2 Formalizing customer security requirements

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

- Incomplete (not all the requirements are present): E.g. important threats that the product should counter are missing;
- 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;
- Implicit: Some product requirements have consequences, but these consequences are themselves not included. It is questionable whether the developer will take these implicit requirements into account;
- Not testable: The requirements are phrased ambiguously, so that it is not possible to verify whether a product meets the requirement or not;
- 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;
- Filled with ambiguous terms like "the communication shall be secure" without defining what "secure" means;
- Inconsistent: The requirements are contradictory.

31 Providing these informal customer requirements to a developer will generally lead to problems, as the developer may misunderstand them. Security evaluation of the resulting product may lead to even more problems, since evaluators may interpret requirements different from both the customer and the developer.

32 For these reasons, an important step in the whole specification-based purchasing process is the formalizing of customer requirements. In the Common Criteria, this formalization takes place using the so-called Protection Profile. A Protection Profile is in essence a document that defines the customer security requirements in a standardized way.

2.3.3 Protection Profiles

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

34 A Common Criteria Protection Profile contains many sections, but for the purpose of this discussion, a very important section is the one on "security functional

requirements (SFR[2])". In the CC, it is mandatory to write these requirements in a special language, defined by the Common Criteria. The usage of this language ensures that the Protection Profile is:

- Not ambiguous: The language contains well defined terms, so that a developer can understand the requirements and interpret them correctly;
- 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 Protection Profile;
- Not too detailed: The language enforces a certain level of abstraction. This closely follows what should be the customer requirements: The customer wants something to be done but does not want to worry how this is accomplished;
- More complete: The language contains several constructions ("if this functionality is required than this other functionality is also required") to help ensure that implicit requirements are included.

35 The process of formalizing customer requirements into a PP is shown in figure 1.

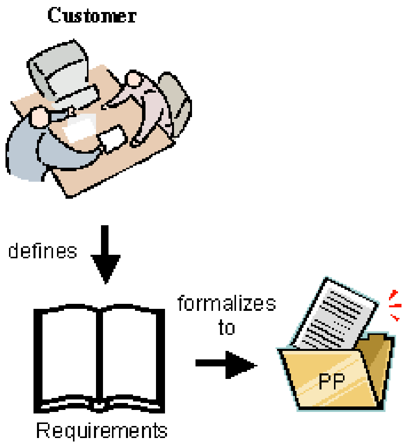

Figure 1 - Formalizing customer requirements into a PP

2.3.4 Building a product from a PP

36 The customer can now give the Protection Profile, i.e. his formalized requirements, to a developer. The developer uses this Protection Profile as a starting point for the development of a product. As a first step in this process he writes a Security Target.

37 A Security Target is very similar to a Protection Profile, but where a PP defines the customer requirements and is in principle written by the customer, the Security Target is a product specification and written by the developer.

[2] Note that SFRs (security functional requirements) are the semi formal language provided by the CC in order to describe security features.

38 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 describe 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 e.g. the product is made for several customers, each with different requirements, or because the product is derived from a standard product.
- The ST contains more detail than the PP: While the PP explains "what" shall be secured, the ST contains additional details about "how": The developer points out, in general terms, how he will implement the customer requirements.

39 The ST defines for the developer which security functionality his product should deliver and serves as security requirements for the rest of the development process of the developer. The process of converting a PP into an ST is shown in figure 2.

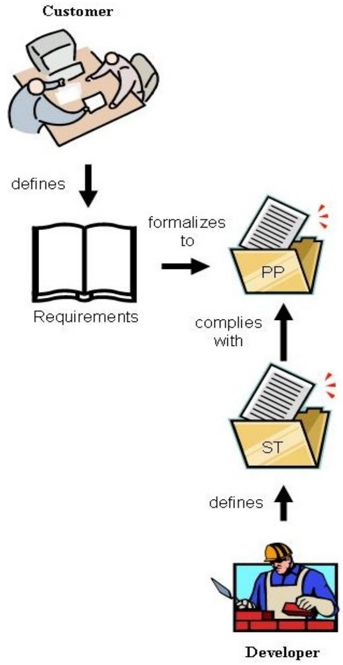

Figure 2 - Converting a PP into an ST

40 Ideally the developer takes the ST as a starting point for the security part of his development process, but often this ST comes in when the development process is already running. In either case, the result of the development process is 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. The complete process is illustrated in figure 3.

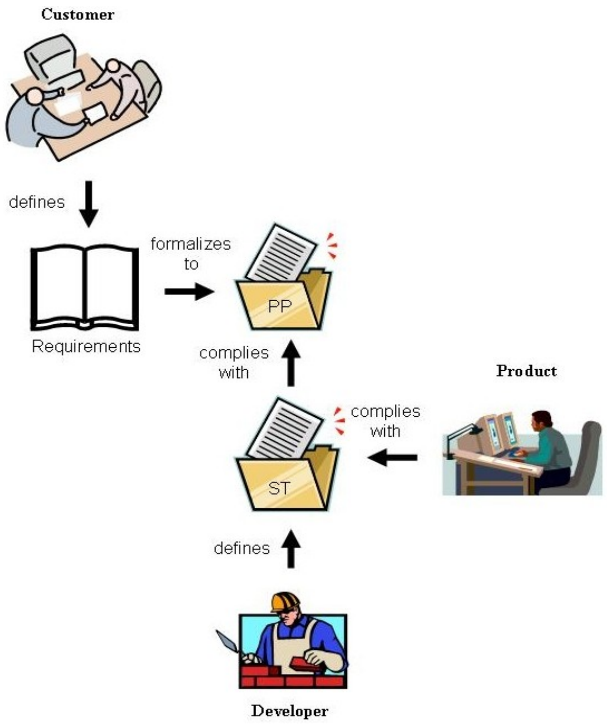

Figure 3 - From a PP to a product in the field

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

41 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):

- My Security Target complies with your Protection Profile.
- My product complies with my Security Target.
- Therefore, my product complies with your Protection Profile.

42 If the customer believes the developer, the process ends here.

43 Otherwise, if the customer prefers an independent verification of these claims, he can enlist a third party (an evaluation facility) to check these claims of compliance (i.e. perform a CC-evaluation). In this process, an evaluation facility can use the PP, the ST, the product and the CC to evaluate two statements:

- The Security Target complies with the Protection Profile.
- The product complies with the Security Target.

44 This evaluation results in a confirmation as illustrated in figure 4.

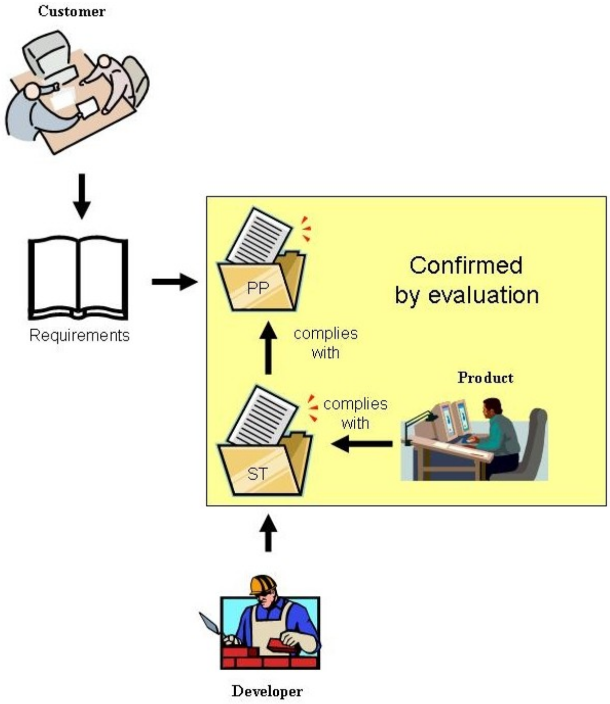

Figure 4 - The role of evaluation in a specification-based purchasing process

45 Note that using evaluated products does not solve everything. In particular two things are left open:

- *The determination and interpretation of the customer's informal security requirements*. As said earlier, this process falls outside the scope of the Common Criteria, but if this is not done correctly, the Protection Profile will not match the customer's requirements and therefore the product will likely not match the customer's requirements either. Note that this guide, and in particular chapter 4, provides assistance in writing a PP.
- *The "confirmation" provided by the evaluation is not absolute*. A CC-evaluation will never provide an absolute guarantee that the product meets the Protection Profile, but it can deliver a certain amount of assurance that the product meets all requirements of the Protection Profile. More on this difficult subject can be found in section 3.5.2.

2.4 Selection-based purchasing processes

46 The previous section discussed a customer delivering a specification and a developer implementing that specification. This section discusses the situation in which the 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 the formalized customer requirements (a Protection Profile), but on the comparison of existing products by the customer.

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

- A developer must produce a product and a specification of this product and provide the specification to the customer;
- 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.

48 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. Each step is discussed in the following sections.

2.4.1 Providing a specification by the developer

49 With specification-based purchasing processes, the customer has to provide a specification to the developer, with selection-based purchasing processes, the developer has to provide a specification to the customer.

50 If this specification is informal, the same potential disadvantages hold as for the informal customer requirements discussed in section 2.3.2: Incomplete, not embedded, implicit, not testable, too detailed, ambiguous and/or inconsistent. For this reason, this specification needs to be formalized as well. For this the CC uses a construct for the specification: The Security Target which was already discussed in section 2.3.4. The ST here is identical to the Security Target discussed in section 2.3.4 with one obvious difference: Since it is not based on a customer's Protection Profile it does not have to comply to a Protection Profile. It may, however, comply to Protection Profiles written by other parties.

51 Because the developer does not know the customer requirements, he will have to make an estimate of these requirements and codify them in the Security Target. This estimate does therefore not necessarily match with the customer requirements.

52 The developer builds his product according to the Security Target: This process is similar to that described for specification-based purchasing processes. This whole process is illustrated in figure 5.

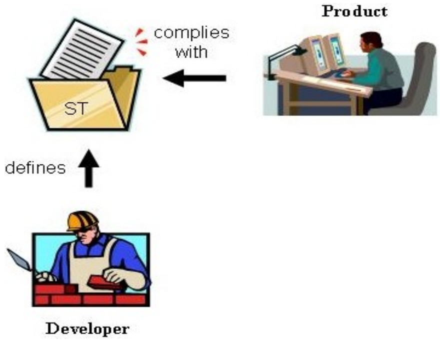

Figure 5 - Writing an ST and producing a product

53 Note: Alternatively it is possible that the developer has an existing product and writes an ST in order to describe the security properties of this product. However the resulting process is the same: The product needs to comply to the ST and the customer can use the ST in order to compare it to his security needs. Therefore this alternative is covered by this chapter.

2.4.2 Comparing Security Targets by the customer

54 The customer can now compare the Security Targets of a number of products and select the best one. This means that he will somehow have to find out what his informal security requirements are (see section 2.3.1) and compare these with the Security Targets offered to him. If one or more products match he has succeeded, if this is not the case he will either have to choose the "closest" product or find some other solution. This whole process is illustrated in figure 6.

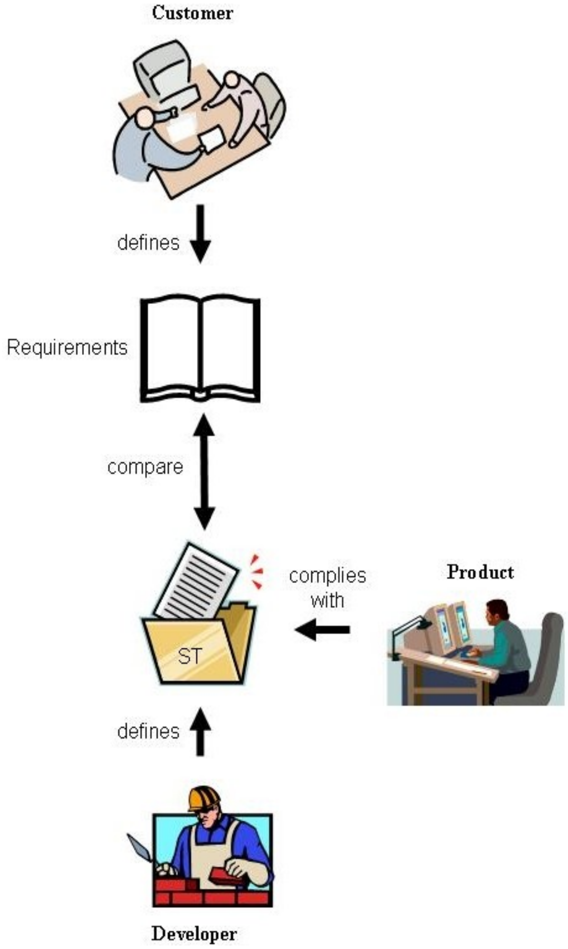

Figure 6 - Selecting a product by comparing STs

55 As already stated in section 2.3.1, the process of deriving informal customer security requirements falls outside the scope of the CC and of this guide. The process of comparing the informal requirements with an ST also falls outside the scope of the CC, but guidance on this topic can be found in section 3.

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

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

57 However, the developer may want to provide an independent verification of this claim. For this, he can commission a third party (an evaluation facility) to check these claims (i.e. perform a CC-evaluation). Alternatively, also the customer may commission a CC-evaluation. In both cases a successful evaluation will result in a certificate confirming the developer's claims.

58 In the Common Criteria, an evaluation facility can evaluate the statement:

- The product meets the Security Target.

59 This confirmation provided by an evaluation is illustrated in figure 7.

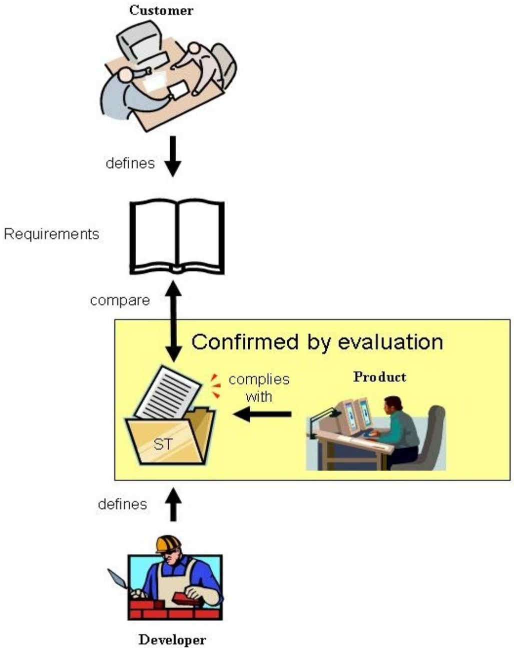

Figure 7 - The role of evaluation in a selection-based purchasing process

60 Note that using evaluated products does not solve everything. In particular two things are left open:

- *The comparison of the customer's informal security requirements with the Security Target*. As said earlier, this process falls outside the scope of the Common Criteria, but if this is not done correctly, the Security Target will not match the customer's requirements and therefore the product will likely not match the customer's requirements either. Note that this guide, and in particular chapter 3, provides assistance in reading and understanding Security Targets.

- *The "confirmation" provided by the evaluation is not perfect.* A CC-evaluation will never provide a perfect guarantee that the product meets the Security Target but it can deliver a certain amount of assurance that the product meets the Security Target. More on this difficult subject can be found in section 3.5.2.

3 Reading Protection Profiles and Security Targets

3.1 Introduction

61 This chapter is targeted specifically at people, who have no deeper knowledge of the Common Criteria, but plan to read a Security Target or a Protection Profile in order to understand, if a product fulfilling this PP or ST would be suitable for their security needs.

62 Readers, who need a more in depth understanding of an ST or PP, for example developers or security evaluators of products, cannot rely on this chapter alone. This audience should use the definitions given in the relevant sections of the CC Parts 1, 2 and 3, but may also benefit from the next chapters, which discuss writing of PPs and STs.

63 This chapter only discusses Security Targets in depth. The reason for this is that the contents required for a Protection profile are mainly a subset of those required for a Security Target. Therefore, if you need to read a Protection Profile, you can use the guidance given for Security Targets.

64 Unfortunately, a Common Criteria Security Target cannot be summarized into a single number or a set of simple properties: Security Targets describe a complex set of security properties of an IT product that, if not carefully read, may lead to surprises when purchasing or using the product. On the other hand, some sections in a Security Target (notably the security functional requirements) are difficult to understand for a newcomer to the CC. The coming subsections will therefore explain the key sections of a Security Target: Sections that are relatively easy to understand, but that contain key information to understanding the security properties of the product described by the Security Target.

65 The most relevant and readable sections are:

- The TOE[3] overview,
- The TOE description,
- The security objectives for the operational environment,
- The conformance claim,
- The TOE summary specification (in case of STs).

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

[3] Note that TOE is the abbreviation for "Target of Evaluation", which is the term used in the CC to denote the evaluated product (or part of a product).

3.2 Reading the TOE overview

67 The TOE overview in general is the first section you should read in an ST, as it *"is aimed at potential consumers 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"* according to CC Part 1, section A.4.2. The TOE overview contains three sections of interest:

- Usage and major security features of the TOE,
- TOE type,
- Required non-TOE hardware/software/firmware.

68 In the following three subsections we discuss each of these in turn.

3.2.1 Usage and major security features of a TOE

69 This section is based on CC Part 1, section A.4.2.1.

70 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.

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

72 An example of a section "usage and major security features" is:

73 *"The MauveCorp MauveRAM Database v2.11 is a multi-user database intended to be used in a networked environment. It allows 1024 users to be active simultaneously. It allows password/token and biometric authentication, protects against accidental data corruption, and can roll-back 10.000 transactions. Its audit features are highly configurable, so as to allow detailed audit to be performed for some users and transactions, while protecting the privacy of other users and transactions."*

3.2.2 TOE type

74 This section is based on CC Part 1, section A.4.2.2.

75 The TOE type is a description of the general category of IT products the TOE belongs to (a firewall, a smart card, an operating system etc.). The CC 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:

- 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. Examples include:

  - A TOE of type "ATM-card", which does not have any identification/authentication functionality;
  - A TOE of type "firewall", which does not support protocols that are almost universally used;
  - A TOE of type "operating system", which has no functionality for access control to files.
- 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. Examples include:
  - A PC-operating system, which is unable to function securely unless the PC has no network connection, floppy drive, and CD/DVD-player;
  - A firewall, which is unable to function securely unless all users that can connect through that firewall can be trusted.

76 **Note that this is the only place in a Security Target where these warnings must be explicitly provided.**

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

3.2.3 Required non-TOE hardware/software/firmware

78 This section is based on CC Part 1, section A.4.2.3.

79 The TOE, especially when it is a software 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.

80 The 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 readers of the ST to determine the major hardware/software/firmware components needed to use the TOE.

81 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, price, company policies etc.

3.3 Reading the TOE description

82 This section and its subsections are based on CC Part 1, section A.4.3.

83 An important aspect to understand of CC-evaluations, is that if you read that the well-known product XYZ has been evaluated, this does not mean that all functional

features (or even a majority of functional features) of this product have been evaluated. It well may be the case that only some of its functional features have actually been looked at and the rest were assumed to be turned off during the evaluation.

84 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.

3.3.1 TOE description - physical scope

85 According to the CC *"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."*

86 You should briefly examine this list to see if you see anything odd on it that you would not expect, or whether some parts of the product are missing. If something is not in this list, the evaluation has completely ignored it and assumed it did not exist. If you intend to use parts, which are not in the list, you need to be aware that the evaluation doesn't tell you anything about the security of such parts.

3.3.2 TOE description - logical scope

87 According to the CC *"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."*

88 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 section 3.2.1) but where that discussion was only a few paragraphs, this discussion is more likely to be a little bit longer. 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 well may be that remote management was not evaluated, and hence, remote management should not be turned on if you want to use the evaluated product.

89 It is therefore important to scrutinize this section to determine whether all security-related features that you require were actually evaluated. If they are not, it well may be that you are required not to use that feature if you want the product to function securely.

3.4 Reading the security objectives for the operational environment

90 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 an important server, this TOE needs to be protected against people accessing it with a screwdriver. This could theoretically be solved by the TOE itself, but since servers with a tamper-proof enclosing are rarely used, in general the operational environment should address this, for example by providing a locked secure room for the server.

91 These and similar requirements for the operational environment are described in a Security Target 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 work. A number of examples follow (drawn from CC Part 1 section A.7.2.2):

- The operational environment shall provide a workstation with the OS Inux version 3.01b to execute the TOE on;
- The operational environment shall ensure that all human TOE users receive appropriate training before allowing them to work with the TOE;
- The operational environment of the TOE shall restrict physical access to the TOE to administrative personnel and maintenance personnel accompanied by administrative personnel;
- The operational environment shall ensure the confidentiality of the audit logs generated by the TOE before sending them to the central audit server.

92 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 organization: The TOE will not do it for you. If a single one of these objectives is not met, the TOE will very likely not function securely.

93 It is therefore imperative that you assess these objectives in order to determine whether they are achievable in your organization, and if one of them is not achievable, the TOE may not be suitable for you.

3.5 Reading the conformance claim

94 The conformance claim usually consists of a single sentence of the form "This Security Target claims conformance to:

- CC v3.1 R3 (the version and the revision of the CC that is used);
- Part 2 extended or conformant (both variants are acceptable for a customer);
- Part 3 extended or conformant (both variants are acceptable for a customer);

- A list of *packages* that the TOE claims conformance with. Usually there is only one such package and its called one of EAL1, EAL2, ..., EAL7, often followed by "augmented". These EALs[4] are standard levels of evaluation and are discussed further in section 3.5.2. If there is a different package, you should determine if a justification is given and if this justification seems acceptable for you.

- A list of *Protection Profiles* that the TOE claims conformance with. This is discussed further below.

3.5.1 Conformance to Protection Profiles

95 As already described in chapter 2 Security Targets may or may not claim conformance to Protection Profiles. If they do claim conformance to a Protection Profile, this is listed here. The CC does not allow any form of partial conformance, so if the PP is listed here, the ST has to be completely conformant to that PP.

96 There are two types of conformance, called "strict" and "demonstrable" conformance (see section 4.2.1 of this guide and CC Part 1, annex D, for details). In both cases conformance to a PP means that the ST (and if the ST is of an evaluated product, the product as well) meets all requirements in the PP.

3.5.2 EALs and other assurance issues

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

- A bus ticket with a small amount of "travel budget" on it;

- A credit card with a €10.000 allowance;

- An access card to a top secret military facility.

98 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 rides on the bus. It is not useful to employ the best (and therefore most expensive) smart cards in the field for this.

99 In the second case, and certainly in the third case, a much "better" card may be required, as the consequences of breaking these cards may be much more extensive.

100 In the CC, this quality is called "assurance" and is defined in the so-called security assurance requirements (SAR) in Part 3 of the CC. The CC measures assurance by examining many aspects 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.

[4] Note that EAL is the abbreviation for "Evaluation Assurance Level", which is the term used in the CC to denote predefined assurance packages.

101 The CC formalizes assurance into 6 categories (the so-called "assurance classes" which are further subdivided into 27 sub-categories (the so-called "assurance families"). In each assurance family, the CC allows grading of an evaluation with respect to that assurance family.

102 For example, a product could score in the assurance family developer test coverage (ATE_COV):

- 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;

103 Unfortunately it is almost impossible for a non-expert to interpret a scorecard consisting of individual ratings for all 27 assurance families. To allow non-experts to assess assurance, the CC has 7 predefined ratings, called evaluation assurance levels or EALs. These are called EAL1 to EAL7, with EAL1 the lowest and EAL7 the highest.

3.5.2.1 Evaluation assurance levels

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

105 The EALs are also strictly hierarchical, if EAL n assigns a certain rating to a certain assurance family, then EAL n+1 will assign the same or a higher rating to that assurance family. Therefore EAL n+1 provides strictly more assurance than EAL n.

106 The flip side of the assurance coin is of course cost: Money and time. In the test coverage 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 money and time, and therefore more cost.

107 A listing of each EAL, together with a description of that EAL and a characterization of the assurance which that EAL provides can be found in chapter 8 of CC Part 3.

3.6 Reading the TOE summary specification

108 The final section that is reasonably accessible to a non-CC expert is the TOE summary specification. This section of an ST is meant to explain how the TOE implements the security features. For instance, if the TOE authenticates its users, this section will describe whether this is done by password, by smart card, by iris-

scanning etc. Reading this section will provide a good overview of the technical construction of the TOE and how it provides its security.

109 In all likelihood, this section will contain references to security functional requirements. These look like FIA_UID.1 (three letters of which the first is an F, followed by an underscore, three more letters, a period and a number). At this level of understanding it is best to simply ignore these and concentrate on the descriptions.

3.7 Summary

110 In summary, this chapter was intended to convey two things:

- That an ST can be reasonably understood from reading a number of sections;
- That these sections may contain important caveats and are therefore vital to understanding the limitations of the evaluation.

111 In the past there have been cases where procurement agencies stated merely that they wanted to purchase a firewall evaluated according to EAL4, for example. Hopefully this chapter has conveyed that a CC-certified EAL4 firewall may have limitations that make it totally unusable for you, and will not provide any relevant security. Depending on the logical scope defined for the TOE or the objectives for the intended operational environment stated in the ST, the product may not provide the security functionality expected by a specific user. This holds regardless of the evaluation assurance level.

3.8 Further reading

112 The sections described in this chapter are the most basic sections of the ST, which can be read by relative laymen and which may be useful in order to decide, whether a product is suitable for the needs of a customer.

113 The other sections in the ST can be commented as follows:

- Reading the various rationales is not necessary for the decision, whether the product is suitable for a customer. Their validity was examined during evaluation of the ST, so the customer can assume their correctness. However, for interested readers they may provide a valuable additional understanding of the overall security context.
- The security functional requirements and the extended components definition are very formalised and contain CC specific terminology. Therefore reading these chapters is usually neither recommended for laymen nor necessary for the determination of the suitability of a product for a customer.
- Reading the security problem definition (SPD) is not strictly necessary for the decision, whether a product is suitable in a certain operational environment, because the security objectives for the operational environment state the necessary conditions for this, while the TOE description shows the provided functionality. However, an interested customer may use the security

problem definition in order to compare, if for example the threats described in the ST are similar to his security concerns and are therefore fitting to his own security needs.

114 Readers, who are interested in these sections of an ST, may find more useful information about their content in later chapters of this guide. Although those chapters are designed for writers of PPs/STs, they may also help readers, who seek deeper understanding.

4 Writing a Protection Profile

115 Authors, who plan to write PPs and STs need an understanding of the structure and content of PPs and STs that goes into much greater detail than the simple reader's guide in chapter 3. However, in CC v3, this is discussed at significant length in CC Part 1.

116 To prevent duplication or inconsistency, we have not repeated this material in this guide, but, in the remainder of this guide, we assume that the reader is familiar with the relevant sections and annexes of CC Part 1, and also has some knowledge of CC Part 3 (in particular chapters APE and ASE) and the CEM[5].

117 It is noted again, that this guide is not meant as a substitution for reading the CC but is intended to give additional guidance based on the contents of the CC.

118 This chapter provides instructions on how to write a Protection Profile. This will be done for each section of a PP individually (see sub-chapter 4.2 ff).

119 Note, that nearly all of the text in this chapter also holds for STs. Therefore chapter 5, which describes, how to write an ST, will refer to this chapter and only discuss modifications and additions necessary for STs. This implies that readers interested in writing an ST may want to have a short look into chapter 5 first, but will also need to use this chapter.

120 In order to prepare the guidance for the individual sections of a PP, we first discuss possible strategies for PP development and derive a strong recommendation for one of these strategies, the "explanation method".

121 In order to prepare this discussion we remind the reader of the following fact: From a security evaluation point of view, understanding the connection between the three PP sections

- "security problem definition",
- "security objectives", and
- "security requirements"

is crucial. Therefore the method used to choose security objectives in order to address the security problem definition and the method used to choose security requirements in order to address the security objectives for the TOE strongly determines the readability and usability of a PP (or ST).

122 We describe two distinct methods for the construction of these three sections:

- The so-called "backward method”. While this method may lead to formally correct PPs, the resulting PPs are not that useful, because they do not really

[5] Note that CEM is the abbreviation for the "Common Methodology for Information Technology Security Evaluation" defined for the CC. It contains instructions how to evaluate the TOE according to the CC.

explain, how a given security problem was solved. This method is therefore only discussed as an example, how a PP should not be developed and as contrast with the other method.

- The method recommended in this guide is called the "explanation method”. This method is based more on analysis and explanation than the "backward method”. This method takes more effort to write, but, if applied correctly, is more likely to be understood by the reader. Since a PP is written once and read many times, we believe this to be a correct trade-off.

123 Note that CC 3.1 distinguishes between two versions of PPs/STs: PPs/STs for EAL1 (low assurance PPs/STs) and PPs/STs for EAL2 and higher. For detailed information please refer to CC, Part 1, annexes A.12 and B.11.

124 Note that this guide does not apply to low assurance PPs/STs. Low assurance PPs/STs do not have a security problem definition, security objectives for the TOE, or rationales. As a consequence of this, there is no systematic development of the contents of the remaining sections, but they are simply stated. Therefore the examples given in this guide may also be useful for low assurance PPs (or STs), but the systematic way of deriving them is not necessary in that case.

4.1 Discussion of the backward method

125 As said above we describe this method only in order to give a negative example, how a formally correct, but not very useful and understandable PP could be produced.

126 The backward method is aimed at minimizing the effort in writing a PP, and especially in writing the rationales. To reach this goal, the main aim is to make all sections as identical as possible. In overview, the backward method consists of the following steps:

- Determine which SFRs and security objectives for the operational environment are desired;

- Create a single security objective for the TOE for each SFR;

- Create an organisational security policy (OSP) for each security objective for the TOE;

- Create an assumption for each security objective for the operational environment;

- The rest (PP introduction and conformance claims).

127 These steps are discussed further below:

128 *Step 1: Determine which SFRs and security objectives for the operational environment are desired.* This is basically done by experience. After writing a number of PPs, you can often do a reasonable job at this. Given the nature of the backward method, if you forget something this is no problem as far as meeting the

APE requirements is concerned, though it may lead to problems later on in the evaluation.

129 *Step 2: Create a single security objective for the TOE for each SFR.* The simplest way is to basically reformulate the SFR in natural language. You then write a security requirements rationale for each security objective for the TOE stating that as the SFR is identical to the security objective the security objective is met.

130 *Step 3: Create an OSP for each security objective for the TOE.* The simplest way is simply restate the security objective as an OSP. You then include a security objectives rationale for each OSP, stating that as the corresponding security objective for the TOE is identical to the OSP, the OSP is met.

131 *Step 4: Create an assumption for each security objective for the operational environment.* By now the formula should be apparent: Each assumption will simply be a restatement of the security objective for the operational environment, and the security objectives rationale will simply state this.

132 *Step 5: The rest:* To finalize the PP, the PP introduction and conformance claims are written.

133 The backward method makes use of the fact that the security problem definition is identical to the later layers, and that "deriving the security problem definition falls outside the scope of the CC". So it states whatever it wants to state, and simply copies this to lower layers. By doing this, it manages to completely avoid the question "Why does this TOE have to do what is in the PP", and simply moves to a statement "This is what the TOE has to do", which is then repeated three times (in the security problem definition, the objectives and the requirements) because the CC says so.

134 This method has two "advantages": It saves work and the resulting PP almost automatically meets many of the requirements for a successful evaluation of the PP.

135 The method has two disadvantages: The resulting PP will probably be almost completely unintelligible to the average reader and will give no motivation, why the security functionality provided by the TOE is useful to address a real security problem. As said above, it is simply a statement "This is what the TOE has to do", and it does not show in any way why the TOE has to do this, how the TOE reflects the security needs of the customer, what the intent of the PP is, etc.

136 These disadvantages more than outweigh the advantages and the method should therefore not be used where it can be avoided. A PP is hopefully used in multiple evaluations, and should therefore be understandable to multiple audiences. Spending a little bit more effort to do so will go a long way in assisting those audiences.

137 For this reason, this guide will use a method that is focused more on explanation. This method is described in the following sections.

4.2 The explanation method

138 In the following sub-chapters we will describe the "explanation method", a method that focuses on deriving the various items in a PP, rather than simply stating them. This method will take more effort to apply, but, if applied correctly, will be easier to understand by the reader. Since a PP is written once and read many times, we believe this to be a correct trade-off.

139 We like to stress that this is just one possible method, and that the APE criteria[6] and the CEM are the sole measure for correctness for a PP. That is, any PP that meets these is a correct PP, regardless of how it is derived. So other methods (including the backward method) to write PPs are allowed.

140 The explanation method consists of the following steps:

- Writing the conformance claims;
- Determining the security problem definition;
- Deriving the security objectives for the TOE and the operational environment including the security objectives rationale;
- Deriving the SFRs including the security requirements rationale;
- Defining the SARs and explain why you have chosen them;
- Writing the PP introduction.

Note that we recommend to write the PP introduction in the end, although it is the first section in the PP. The reason is that the process of collecting all the information necessary for the other sections is the best method to enable the author to formulate an introduction.

Each of these steps will be discussed in the following sections.

141 We will illustrate several steps of the explanation method using a simple example: A firewall TOE. We will use this example throughout this chapter as it is both simple enough to be understood, yet complex enough to showcase many aspects of producing a PP.

4.2.1 Step 1: Writing the conformance claims

142 The conformance claims section of a PP describes how the PP (and the resulting TOE) conform to:

- *The CC*: This consists of listing the exact version of the CC that was used to write (and presumably evaluate) the PP. If a translation of the CC was used, this should also be indicated. If any international or national interpretations or supporting documents are used, these should be listed as well.

[6] Note that APE is the acronym for the assurance class for Protection Profile evaluation.

- *Part 2 and 3:* Describing the specific conformance to CC Part 2 and CC Part 3 can be done as followed:

  1) CC Part 2 conformant - a PP or ST is CC Part 2 conformant if all SFRs in that PP or ST are based only upon functional components in CC Part 2, or

  2) CC Part 2 extended - a PP or ST is CC Part 2 extended if at least one SFR in that PP or ST is not based upon functional components in CC Part 2.

  3) CC Part 3 conformant - a PP or ST is CC Part 3 conformant if all SARs in that PP or ST are based only upon assurance components in CC Part 3, or

  4) CC Part 3 extended - a PP or ST is CC Part 3 extended if at least one SAR in that PP or ST is not based upon assurance components in CC Part 3.

- *Protection Profiles*: This consists of listing the Protection Profiles that this PP claims conformance to (if any). Usually a simple list suffices and no extra information is needed in this section.

  Note: If however, the PP claims strict conformance to a PP, which allows demonstrable conformance, this should be identified explicitly. Otherwise the reader (and the evaluator of the PP) would assume that a PP is claimed according to the type of conformance defined in that PP. The opposite case is not possible: If a claimed PP requires strict conformance, a claim of demonstrable conformance is not allowed.

- *Packages*: This consists of listing the assurance package that is used (normally an EAL plus any augmentations). If, by the time you are reading this, other assurance packages than the EALs have been introduced, or functional packages have been defined, and you conform to them, you may list them here as well. Again: A simple list suffices: No extra information is need in this section.

143 Note: If you follow this guide and in particular the recommendations in chapter 4.2.5, you may only be able to determine the assurance package during step 5 of the "explanation method" as described in that chapter. Therefore it is recommended to leave this portion of the conformance claim open until that step is completed.

144 The PP must also describe how other PPs and STs shall conform to the PP. There are two choices for this:

- Strict: Conceptually, this means that the conforming PP/ST must contain everything in this PP. See CC Part 1 annex D for a more exact definition.

- Demonstrable: Conceptually, this means that the conforming PP/ST must be "similar" to this PP. See CC Part 1 annex D for a more exact definition.

Unless you are an expert, use the following guidelines:

- If you want to control the security functionality provided by conforming products very strongly, use "strict". This for example may be the case, if your PP is for a very specific TOE (the intranet for the XYZ army base) or you are the first to write a PP for a specific technology.
- If you only want to be sure that conforming products give an adequate level of protection for your assets, but without prescribing the detailed functionality, use "demonstrable". This for example may be the case if you write a PP for types of products, where developers already have experience with evaluations and you want to allow re-use of existing evaluations. (In that situation "strict" conformance might force a developer to re-evaluate a product using different SFRs in spite of already having a successful evaluation for a similar security problem). This is especially the case, if you write a PP when there are already many PPs for that technology (smart cards, firewalls).

145 If you claim conformance to a functional package or a PP, much of the threats/OSPs/assumptions, security objectives and security requirements may already be taken from that package or PP, and you are advised to make sure that any additional contents for those sections defined by you need to be consistent to those predefined contents.

4.2.1.1 Running example: Writing the conformance claims

146 Typical conformance claims could be defined as follows:

CC conformance claims

- This PP has been developed using version 3.1 R2 of Common Criteria (CC).
- This PP is conform to Part 2 and 3 of the CC; no extended components have been defined.

PP claim

- This PP does not claim conformance to any other Protection Profile.

Package claim

- This PP claims assurance package EAL3 as defined in Common Criteria Part 3.

Conformance statement

- The PP requires strict conformance of any PPs/STs to this PP.

147 The only choice in this list, which needs explanation, would be that for assurance level EAL3. The choice of assurance requirements will be discussed in chapter 4.2.5.

4.2.2 Step 2: Determining the security problem definition

4.2.2.1 OSPs

148 CC Part 1, section A.6.1. (which addresses STs, but is equally valid for PPs) states that *"The security problem definition defines the security problem that is to be addressed. The security problem definition is, as far as the CC is concerned, axiomatic. That is, the process of deriving the security problem definition falls outside the scope of the CC. However, it should be noted that 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."*

149 The explanation method takes this statement very seriously. A PP will be more useful if its readers can understand what the PP contains and how it relates to their situation. An important part of this may be realized by the OSPs.

150 The CC, Part 1, section A.6.3 describes OSPs as *"security rules, procedures or guidelines imposed (or presumed to be imposed) now and/or in the future by an actual or hypothetical organisation in the operational environment. OSPs may be laid down by the organisation controlling the operational environment of the TOE, or they may be laid down by legislative or regulatory bodies. OSPs can apply to the TOE and/or the operational environment of the TOE."* For PPs, the rules laid down by legislative or regulatory bodies will probably be the most important sources for OSPs. In addition, the security policies defined by "Information Security Management Systems (ISMS)" of an organisation can be important for the products purchased by that organisation. If relevant parts of laws, regulations etc. are copied ad verbatim into a PP as OSPs, it can be demonstrated to those legislative or regulatory bodies that the PP is compliant with the laws or rules that they are laying down.

151 A good example is the European Union's Electronic Signature Directive, which contains the following rules for secure signature-creation devices:

- Secure signature-creation devices must, by appropriate technical and operational means, ensure at the least that:

  1) The signature-creation-data used for signature-creation can practically occur only once, and that their secrecy is reasonably assured;

  2) The signature-creation-data used for signature-creation cannot, with reasonable assurance, be derived and the signature is protected against forgery using currently available technology;

  3) The signature-creation-data used for signature-creation can be reliably protected by the legitimate signatory against the use of others.

- Secure signature-creation devices must not alter the data to be signed or prevent such data from being presented to the signatory prior to the signature process.

152 Another example may come from national security rules, such as:

- All products used to store and/or process data classified up to “Secret” must use the National Standard for Password Generation and Encryption [PurpleGoldBook].

- If a system contains classified data, all external repair personnel of that system shall be accompanied by a system administrator with sufficient clearance for that system.

153 When writing a PP you should therefore examine all applicable laws and regulations for items that can be used as OSPs. You should try to cover all security related laws and regulations relevant for the TOE, as this will allow you to show that your TOE actually conforms to those rules.

154 You should then copy the relevant parts of these rules as OSPs into your PP, thus creating a demonstrable link between the laws and regulations and your PP.

155 Note that OSPs of this type are usually informal and/or vague, as the underlying law/regulation is unlikely to use a rigorous framework to define security needs. As these OSPs will be interpreted in a more rigorous way in the rest of the PP, by translating them to security objectives and security requirements this should not pose problems at this stage.

156 In the remainder of this chapter we will derive a PP for a firewall as a running example to illustrate a number of concepts. This example does not use OSPs (it is allowed to use OSPs in a firewall PP, this PP example simply does not do so).

4.2.2.2 Threats

157 CC Part 1, section 7.1 states that *"Security is concerned with the protection of assets. Assets are entities that someone places value upon. Examples of assets include:*

- *contents of a file or a server;*

- *the authenticity of votes cast in an election;*

- *the availability of an electronic commerce process;*

- *the ability to use an expensive printer;*

- *access to a classified facility.”*

158 It is important to realize that the TOE itself (including its mechanisms) is NOT an asset as far as threats are concerned; the threats should be reasons for using the TOE, not results of properties of the TOE itself.

159 CC Part 1, section 7.1 goes on to state *"The environment(s) in which these assets are located is called the operational environment. Examples of (aspects of) operational environments are:*

- *the computer room of a bank;*
- *a computer network connected to the Internet;*
- *a LAN;*
- *a general office environment."*

160 These operational environments can therefore have physical aspects (in the computer room of a bank) and/or logical aspects (connected to the internet).

161 CC Part 1, Section A.6.2, subsequently defines threats as *"A threat consists of an adverse action performed by a threat agent on an asset.*

162 *Adverse actions are actions performed by a threat agent on an asset. These actions influence one or more properties of an asset from which that asset derives its value.*

163 *Threat agents may be described as individual entities, but in some cases it may be better to describe them as types of entities, groups of entities etc.*

164 *Examples of threat agents are hackers, users, computer processes, and accidents. Threat agents may be further described by aspects such as expertise, resources, opportunity and motivation."*

165 To define the threats, you have to answer the question: "What happens when I don't have a TOE (or when I have a TOE that provides no security at all)"? This question can be subdivided into three sub-questions:

- What are the assets? (Informally: What are the items that a prospective TOE owner would find so important that he would want to buy a TOE for their protection?)
- What are the adverse actions? (Informally: Why do these assets have value, and how could this value be diminished?)

  Note: The discussion of the security problem definition is on the level of the question "What happens when I don't have a TOE?" Therefore, the "adverse actions" can not be on the level of attacks against the TOE, which are discussed in a vulnerability analysis of the TOE, for example.
- Who are the threat agents? (Informally: Who or what would cause this value to diminish?)

4.2.2.3 Assumptions

166 The explanation method does not use assumptions extensively. The reason for this is that assumptions are often used as a form of the backward method (see section 4.1) by restating security objectives of the operational environment back to the SPD.

167 The following example shows, how assumptions should **not** be used:

168 Some security functionality of the TOE, for example an authentication method using a password, needs support by the operational environment, which can be identified as a security objective for the environment, e.g.

- OE.RESPONSIBLE_PASSWORD_HANDLING: Users of the TOE will handle their passwords responsibly and particular will not disclose them to unauthorised persons.

169 In the backward method this would simply be re-translated into an assumption with identical content, e.g.:

- A.RESPONSIBLE_PASSWORD_HANDLING: Users of the TOE will handle their passwords responsibly and in particular will not disclose them to unauthorised persons.

170 However, this is not allowed in the explanation method and it is not necessary because the objective for the environment can be traced to the same threat (or OSP), which gave the reason to require an authentication functionality.

171 An example for a legitimate assumption might exist in the following case: A company having physically secure computing centres for their data processing plans to write a PP for a security component (e.g. a firewall) to be used in these computing centres. In this case it is known in advance that physical access to the systems will not be possible for attackers and that therefore physical attacks are not a threat. In this case the following assumption can be stated:

- A.SECURE_LOCATION: The TOE will be located in a physically secure building.

172 However, the same assumption would not be a legitimate one in the sense of the explanation method, if it had been concluded backwards from a property of the TOE (for example: The TOE is a software product and therefore needs physical protection). In this case the corresponding security objective for the environment should be derived from a threat or OSP and not from an assumption.

173 As stated for threats earlier, the explanation method requires the SPD to be written with the following question in mind: "What happens when I don't have a TOE (or when I have a TOE that provides no security at all)".

174 Therefore only assumptions are acceptable in the explanation method, in cases, where certain properties of the TOE environment are already known when deriving the SPD, but not when they are derived from specific properties of the TOE (i.e. a

specific identification and authentication (I&A) mechanism realised by a user-name/password combination).

175 Assumptions are therefore rarely used in the explanation method.

4.2.2.4 Running example: Determining the security problem definition

176 We will analyse the questions from chapter 4.2.2.2 for a simple example: A firewall TOE. We will use this example throughout this chapter as it is both simple enough to be understood, yet complex enough to showcase many aspects of producing a PP. The running example is only based on threats. It does not make use of OSPs and assumptions.

177 So the main question becomes: What happens if you have no firewall? Usually a firewall is used to connect a Local Area Network (LAN) to a Wide Area Network (WAN), usually the internet. What would happen if you connect a LAN directly to the Internet?

178 In this modern day and age, in very little time, all data on your LAN would start disappearing or be corrupted, and the workstations connected to that LAN would stop working. From this we can answer the three sub-questions:

- What are the assets? Apparently the data on your LAN and the IT processes on the LAN.
- What are the adverse actions? Stealing/corrupting the data and impeding the IT processes on the LAN.
- Who are the threat agents? These are harder to pinpoint: Worms, viruses, hackers, etc. We can summarize them as "entities on the WAN (e.g. worms, hackers etc.)"

And from this we come to the following threats relevant to the firewall:

- T.LEAK_TO_WAN: An entity on the WAN (e.g. worm or hacker) causes confidential data on the LAN to leak to the WAN.
- T.CORRUPT_ON_LAN: An entity on the WAN (e.g. worm or hacker) corrupts data on the LAN.
- T.DEGRADE_LAN: An entity on the WAN (e.g. worm or hacker) causes IT processes on the LAN to be degraded or halted.

179 A second class of threats stems from the users on the LAN itself. Unrestricted access to the WAN (e.g. visiting porn/gambling/leisure sites) may cause productivity losses or damage to the reputation of the owner of the LAN. From this we can answer the three sub-questions:

- What are the assets? Staff productivity and reputation of the owner.

- What are the adverse actions? Decreasing productivity and decreasing reputation.
- Who are the threat agents? Company staff (human users on the LAN).

And from this we come to the following threat relevant to the firewall:

- T.UNRESTRICTED_WAN_ACCESS: A human user on the LAN may decrease productivity or cause reputation loss by accessing certain parts of the WAN.

180 And these four threats constitute the threats and thereby the SPD for our firewall example.

4.2.3 Step 3: Deriving the security objectives

181 In this section we describe how to derive the security objectives for the TOE and the operational environment based on the security problem definition provided in step 2.

182 CC Part 1, section A.7 states that *"The security objectives are a concise and abstract statement of the intended solution to the problem defined by the security problem definition. The role of the security objectives is threefold:*

- *provide a high-level, natural language solution of the problem;*
- *divide this solution into two part wise solutions, that reflect that different entities each have to address a part of the problem;*
- *demonstrate that these part wise solutions form a complete solution to the problem."*

183 The purpose of the security objectives is therefore to provide a solution to the security problem definition, consisting of relatively small "chunks", so that this solution becomes manageable and easy to analyse.

184 The security objectives for the TOE form a bridge between the threats and OSPs on one side, and the SFRs on the other side. The level of abstraction of these objectives should therefore be intermediate between the level of the threats and OSPs, which are written from a potential user's point of view, on the one hand, and the semi-formal model of the SFRs on the other hand. This means that they should not simply restate the threats and OSPs nor should they be a natural language form of the SFRs.

185 As the SFRs are largely implementation-independent, the security objectives for the TOE should be implementation-independent as well: They should state what the TOE should do, and not how it should do it.

186 The security objectives for the operational environment should be at a commensurate level of abstraction to the security objectives for the TOE. Similarly to the security objectives for the TOE, they should state what the operational environment should achieve, and not necessarily how this is to be achieved.

187 There is no real algorithm to derive the security objectives from the security problem definition. Doing this correctly is a question of experience, and will likely require several iterations of security objectives, security requirements and security problem definition before they are done. As a rule of thumb, the explanation method considers at least the following three items:

- Where will the TOE be placed and can it be physically attacked there?
- What is the purpose of the TOE?
- How is the TOE managed?

188 To clarify this rather cryptic list, we continue with the running firewall example from the previous section.

4.2.3.1 Running example: Deriving the security objectives

189 In the running firewall example, we had stated a security problem definition consisting of four threats:

- T.LEAK_TO_WAN: An entity on the WAN (e.g. worm or hacker) causes confidential data on the LAN to leak to the WAN.
- T.CORRUPT_ON_LAN: An entity on the WAN (e.g. worm or hacker) corrupts data on the LAN.
- T.DEGRADE_LAN: An entity on the WAN (e.g. worm or hacker) causes IT processes on the LAN to be degraded or halted.
- T.UNRESTRICTED_WAN_ACCESS: A human user on the LAN may decrease productivity or cause reputation loss by accessing certain parts of the WAN.

190 We now postulate a TOE and an operational environment that will counter these threats, and start answering the three questions posed earlier:

- Where will the TOE be placed and can it be physically attacked there?
- What is the purpose of the TOE?
- How is the TOE managed?

191 It is of crucial importance to note that the answers to these questions do NOT follow from the threats: They are choices to be made by the PP author in countering the threats. Several answers are possible to each question, and each will give rise to a different PP.

Where is the TOE placed?

192 Every TOE must have both a "logical" place and a "physical" place. We start with the "logical" place. The firewall TOE must keep the LAN and the WAN apart, it must therefore be placed between the LAN and the WAN, and there should be no possible bypass of it. As the TOE itself cannot arrange this, this becomes a security objective for the operational environment:

- OE.ONLY_CONNECTION: The operational environment shall ensure that the TOE is placed between the LAN and the WAN in such a way that all network traffic between LAN and WAN must pass through the TOE.

193 Note that we introduced a little bit more detail into this security objective: It now talks about network traffic. This is an example of the "intermediate" level of abstraction that we discussed earlier.

194 For the physical place, we have to consider that each TOE may be physically attacked (by someone taking a screwdriver to it), and that this must somehow be covered: Either by making the TOE itself physically secure (i.e. an armoured firewall, or a redundant TOE in several places etc.) or by the operational environment (placing it in a physically secure room, or by insuring the TOE etc.). Traditionally firewalls are not armoured and kept in a physically secure environment, and this is the solution we choose for our firewall as well:

- OE.PHYSICALLY_SECURE: The operational environment shall ensure that the TOE is physically placed in a room that can only be accessed by trusted personnel.

195 Note that if we wanted, we could have something like “OT.ARMOR_TOE: The TOE shall protect itself against physical attack" to model a TOE that can physically defend itself. Or we could have a more abstract “OE.PHYSICALLY_SECURE: The operational environment shall ensure that physically attacking the TOE is impossible and/or has no effect" to provide much more options to the owner of the operational environment to comply with it: He could now deploy guards, have duplicate redundant elements etc.

What is the purpose of the TOE?

196 If we look at the threats we see WHAT is to be achieved from the potential user's point of view. If we look at the role of the TOE in this, and add somewhat more detail as to what the TOE contributes to this, we can describe the main purpose of the TOE as twofold.

- The TOE should examine network traffic from the WAN to the LAN, and only allow network traffic to pass through that cannot cause confidential data to be leaked outside, corrupt LAN data, or degrade/halt IT processes.

- The TOE should examine network traffic from LAN to WAN and only allow network traffic to pass through that cannot decrease productivity or cause reputation loss.

197 However, these are not suitable to be used as security objectives, since they are quite subjective: Different LAN configurations will react differently to different network traffic. And vice versa, for some organizations, certain WAN access will cause loss of reputation, while for other organizations this will be daily routine. The operational environment must therefore define a policy on which traffic is allowed and which traffic is not allowed:

- OE.TRAFFIC_POLICY: The operational environment shall define a policy on which incoming network traffic is allowed and which outgoing network traffic is to be allowed.

198 The TOE must of course implement this policy, and this leads to the following security objectives:

- OT.OUTGOING_TRAFFIC: The TOE shall be able to block network traffic from LAN to WAN according to the defined traffic policy.

- OT.INCOMING_TRAFFIC: The TOE shall be able to block network traffic from WAN to LAN according to the defined traffic policy.

199 Finally, the operational environment should be able to cope with any incoming traffic that the policy allows. This leads to the following security objective:

- OE.RESIST_REMAINING_TRAFFIC: The operational environment shall ensure that any allowed incoming network traffic cannot cause confidential data from the LAN to be leaked outside, corrupt data on the LAN, or degrade/halt IT processes on the LAN.

200 Note, that there is no need for an equivalent objective for outgoing traffic: The policy regarding incoming traffic is connected to potential harm done to the LAN, so it is up to the environment to ensure that remaining allowed traffic doesn't do any harm to the LAN. On the other hand outgoing traffic is based on rules about what is allowed for users (e.g. the users are not allowed to call certain services), however these rules are not concerned with technical damage done to the WAN or LAN, but are chosen according to non-technical criteria (image loss etc.) and are simply enforced as defined.

201 Together these four security objectives describe how the TOE and the operational environment will solve the main part of the security problem.

How is the TOE managed?

202 As networks change, and policies change and technology changes, it is unlikely that the defined policies for incoming and outgoing network traffic never change. It must therefore be possible to change these policies. If every entity on the WAN or LAN could change these policies, the policies could effectively always be circumvented. It is therefore necessary to define a trusted person "the operator" who is the only one who can change the policies. Or, better defined as security objectives:

- OT.OPERATOR: The TOE shall allow only the operator to change the traffic policy.
- OE.TRUSTED_OPERATOR: The operational environment shall ensure that the operator is adequately trained and can be trusted to perform his duties.

Summarized security objectives

203 From the previous paragraphs we can now gather the security objectives and summarize them here.

204 The security objectives for the TOE:

- OT.OUTGOING_TRAFFIC: The TOE shall be able to block network traffic from LAN to WAN according to the defined traffic policy.
- OT.INCOMING_TRAFFIC: The TOE shall be able to block network traffic from WAN to LAN according to the defined traffic policy.
- OT.OPERATOR: The TOE shall allow only the operator to change the defined traffic policy.

205 The security objectives for the operational environment:

- OE.TRAFFIC_POLICY: The operational environment shall define a policy on which incoming network traffic is allowed and which outgoing network traffic is to be allowed.
- OE.ONLY_CONNECTION: The operational environment shall ensure that the TOE is placed between the LAN and the WAN in such a way that all network traffic between LAN and WAN must pass through the TOE.
- OE.PHYSICALLY_SECURE: The operational environment shall ensure that the TOE is physically placed in a room that can only be accessed by trusted personnel.
- OE.RESIST_REMAINING_TRAFFIC: The operational environment shall ensure that any allowed incoming network traffic cannot cause confidential data from the LAN to be leaked outside, corrupt data on the LAN, or degrade/halt IT processes on the LAN.
- OE.TRUSTED_OPERATOR: The operational environment shall ensure that the operator is adequately trained and can be trusted to perform his duties.

206 Note that for this example we derived objectives from threats only. PPs/STs in general can also contain OSPs and assumptions. The method demonstrated by the example is the same for these parts of the SPD, with the additional constraint, that assumptions can only be realised by objectives for the environment.

4.2.3.2 Writing the security objectives rationale

207 As stated earlier, CC Part 1, section A.7 states that *"The security objectives are a concise and abstract statement of the intended solution to the problem defined by the security problem definition. The role of the security objectives is threefold:*

- *provide a high-level, natural language solution of the problem;*
- *divide this solution into two part wise solutions, that reflect that different entities each have to address a part of the problem;*
- *demonstrate that these part wise solutions form a complete solution to the problem."*

208 The first two items of this list have been handled in the previous sections. The third item of this list is to be handled by the security objectives rationale.

209 CC Part 1 section A.7.3 states that *"a security objectives rationale [contains] two sections:*

- *a tracing that shows which security objectives address which threats, OSPs and assumptions;*
- *a set of justifications that shows that all threats, OSPs, and assumptions are effectively addressed by the security objectives."*

210 The first one is best addressed by a table, having the threats, OSPs and assumptions on the rows, and the security objectives on the column, and "X" marking a place in the table if that security objective traces to that threat, OSP or assumption. Its purpose is to show that:

- Each security objective traces to at least one threat, OSP or assumption. In other words, each column has to have at least one "X" in it.
- Each threat, OSP and assumption has at least one security objective tracing to it. In other word, each row has to have at least one "X" in it.
- Assumptions can only be traced to by security objectives for the operational environment.

211 If you have been using the explanation method, the tracing can easily be deduced from the analysis you did to get to the security objectives. The result is given in table 1 (see the following chapter).

212 The second one is slightly harder. For each threat and OSP, you have to describe HOW the security objectives counter/meet it. If you have done the analysis prescribed by the explanation method, this can be derived from that analysis, as all the elements that are needed are there. They just need to be collected and described in a short, to-the-point, and understandable way.

4.2.3.3 Running example: Writing the security objectives rationale

213 The tracing of objectives to the threats (there are no assumptions or OSPs in this example) is given by the following table:

	OT.OUTGOING_TRAFFIC	OT.INCOMING_TRAFFIC	OT.OPERATOR	OE.TRAFFIC_POLICY	OE.ONLY_CONNECTION	OE.PHYSICALLY_SECURE	OE.RESIST_REMAINING_TRAFFIC	OE.TRUSTED_OPERATOR
T.LEAK_TO_WAN		X	X	X	X	X	X	X
T.CORRUPT_ON_LAN		X	X	X	X	X	X	X
T.DEGRADE_LAN		X	X	X	X	X	X	X
T.UNRESTRICTED_WAN_ACCESS	X		X	X	X	X		X

Table 1: Mapping of security objectives and threats

214 The justifications are derived straight forward from the earlier considerations in 4.2.3.1:

215 T.LEAK_TO_WAN, T.CORRUPT_ON_LAN and T.DEGRADE_LAN are countered as follows:

- First it must be determined which network traffic the LAN should be able to resist and which network traffic might cause one of the threats to be realized. As this is different for each LAN, this is done by the operational environment (OE.TRAFFIC_POLICY).
- The operator can then set and maintain the TOE (OT.OPERATOR) to block all network traffic that is considered threatening (OT.INCOMING_TRAFFIC). Of course this only works when all network traffic is actually routed through the TOE (OE.ONLY_CONNECTION) and the operator does his job correctly (OE.TRUSTED_OPERATOR).
- Simultaneously the LAN must be configured (and maintained) to ensure that it actually resists all remaining network traffic (OE.RESIST_REMAINING_TRAFFIC) and that the TOE must be kept safe from physical attack (OE.PHYSICALLY_SECURE) to be able to keep performing its functions.

216 T.UNRESTRICTED_WAN_ACCESS is countered as follows:

- First it must be determined which network traffic to the WAN could cause reputation loss or decrease productivity. As this is different for each organization, this is done by the operational environment (OE.TRAFFIC_POLICY).
- The operator can then set and maintain the TOE (OT.OPERATOR) to block all network traffic that is considered threatening (OT.OUTGOING_TRAFFIC). Of course this only works when all network traffic is actually routed through the TOE (OE.ONLY_CONNECTION) and the operator does his job correctly (OE.TRUSTED_OPERATOR) .
- Finally the TOE must be kept safe from physical attack (OE.PHYSICALLY_SECURE) to be able to keep performing its functions.

217 Note that this example does not use OSPs or assumptions. If there had been OSPs or assumptions, the security objectives would have needed to address these as well.

4.2.3.4 The explanation method

218 From the running example and especially from the security objective rationale, we can see where the explanation method gets its name. Where the backward method creates the SPD "backwards" from the objectives by simply restating them, the explanation method:

- Usually has more security objectives than it has threats and OSPs;
- Usually has security objectives that are qualitatively different than the threats and OSPs (at another level of abstraction);
- Usually has much richer and more complex relations between threats/OSPs and security objectives than the simple 1-1 relations from the backward method and therefore more meaningful rationales to *explain* why these security objectives sufficiently address the threats and OSPs, instead of simply restating those threats and OSPs.

219 And from this, it can be understood why the explanation method uses assumptions only in specific cases. Tracing a security objective for the operational environment back to an assumption does only provide one explanation for that security objective: The security objective is necessary, because the assumption was made, they merely restate each other. It is therefore not allowed in the explanation method, to choose an assumption because it is needed for an objective. Doing that would leave unclear WHY this security objective is necessary and what its relation is with the other security objectives. In the explanation method, the security objectives rationale shows why the security objective for the operational environment is required, therefore only those assumptions are allowed, which are really known in advance and are not derived from specific properties of the TOE.

220 Additionally, using the explanation method will allow analysis of the line of reasoning that is used. An evaluator (or the developer himself) can examine the rationale to look for holes. In the backward method, this is not useful: Each objective has an exact 1-1 correspondence with a single threat/OSP/assumption so there is nothing to examine (except syntactical restatements).

4.2.4 Step 4: Deriving the SFRs

221 In this section we describe how to derive the security functional requirements from the security objectives for the TOE.

222 CC, Part 1, section A.9.1 states that:

223 *"The SFRs are a translation of the security objectives for the TOE. They are usually at a more detailed level of abstraction, but they have to be a complete translation (the security objectives must be completely addressed) and be independent of any specific technical solution (implementation). The CC requires this translation into a standardised language for several reasons:*

- *to provide an exact description of what is to be evaluated. As security objectives for the TOE are usually formulated in natural language, translation into a standardised language enforces a more exact description of the functionality of the TOE.*

- *to allow comparison between two STs. As different ST authors may use different terminology in describing their security objectives, the standardised language enforces using the same terminology and concepts. This allows easy comparison."*

224 A word of caution: Though the term "translation" is used in the cited text, this does not imply a one-to-one relation between all details in the security objectives on the one hand and the SFRs on the other hand. Since the SFRs are usually more detailed, it is not necessary to find every detail of the SFRs in the security objectives, while of course every detail of the security objectives needs to be addressed by the SFRs. A one-to-one correspondence between security objectives and SFRs would be a hint for the use of the "backward"-method, which we do not recommend.

225 The process of specifying the SFRs for a TOE is best split into two sub-steps:

- Sub-step 1: Analyse the objectives for the TOE in order to decide, which thematic groups of SFRs are needed to fulfil the objectives and which specific SFRs in each group may be useful.

- Sub-step 2: Specify the SFRs in detail, defining subjects, objects, operations and other entities required by the SFRs and performing the operations as far as useful for the PP.

4.2.4.1 Sub-step 1: Analysing the security objectives for the TOE

226 A set of possible security functional requirements is defined in the CC, Part 2. There they are grouped in classes and families, however we suggest to choose from them according to slightly different thematic considerations.

Choosing the relevant thematic groups:

227 The classes defined in CC, Part 2, are meant to group the possible SFRs thematically. However in practice some thematic groups can be defined as parts of classes or may even contain families from several classes. We recommend to consider the following thematic groups, when deriving SFRs:

-	Logging and audit	class FAU
-	Identification and authentication	class FIA
-	Cryptographic operation	class FCS
-	Access control	families FDP_ACC, FDP_ACF
-	Information flow control	families FDP_IFC, FDP_IFF
-	Management functions	class FMT
-	(Technical) protection of user data	families FDP_RIP, FDP_ITT, FDP_ROL
-	(Technical) protection of TSF data	class FPT
-	Protection of (user) data during communication with external entities	families FDP_ETC, FDP_ITC, FDP_UCT, FDP_UIT, FDP_DAU, classes FCO and FTP

228 There are some other thematic areas, which are either seldom used today (e.g. class FTA, TOE access) or have a very specific meaning, which should be obvious, when needed (class FPR, privacy).

229 In cases, where authentication is used as pre-condition to be able to enforce access control, it may be useful to group both types of SFRs into one thematic section in the PP.

Checking the dependencies:

230 The formal dependencies between SFRs, as defined in Part 2, give additional hints, which SFRs have strong interconnections and may therefore be suitable for one thematic group. The PP author has to take them into consideration, when collecting SFRs.

Cross-check with unused thematic groups:

231 As an additional cross-check it is advisable to also go the other way around: Go through those of the thematic groups listed above, which have not been chosen in this sub-step, and cross-check, if they seem to be applicable to the TOE.

232 Next we show the derivation process for the running example.

4.2.4.2 Running example: Deriving the SFRs, sub-step 1

233 Analysis of the security objectives for the TOE and choice of thematic groups:

234 In the running example of a firewall we had identified the following objectives for the TOE:

- OT.OUTGOING_TRAFFIC: The TOE shall be able to block network traffic from LAN to WAN according to the defined traffic policy.

- OT.INCOMING_TRAFFIC: The TOE shall be able to block network traffic from WAN to LAN according to the defined traffic policy.

- OT.OPERATOR: The TOE shall allow only the Operator to change the defined traffic policy.

235 The first two objectives look like "information flow", so one can decide to use an information flow policy here (components from FDP_IFC, FDP_IFF). The third objective concerns management of the TOE, so management SFRs from class FMT seem suitable, and it requires to restrict these management functions to a specific role, which can be modelled by the use of identification and authentication from class FIA.

236 Checking the dependencies:

237 A check of the dependencies between the SFRs shows, that there are dependencies from information flow SFRs to management SFRs (from FDP_IFF.1 to FMT_MSA.3). In fact it is clear that management SFRs and the information flow policy are suitable to complement each other: The operator of the firewall is the one, who is allowed to manage the information flow policy. So the dependency check has not revealed new requirements, but supports the model already chosen. (Note: The later detailed specification will show, that this specific management SFR is not needed.)

238 Cross-check of the thematic groups:

239 For some groups it is obvious that they do not apply in our simple example: Cryptographic functions are not needed, therefore class FCS is not relevant. Similarly no additional SFRs related to user data are relevant, because the firewall has no other purpose then managing the flow of data from one network interface to the other. Direct access to user data via other interfaces, storage of user data etc. are not relevant, so other components from class FDP are not needed. Similarly no

specific additional functions for protection of TSF data (class FPT) are needed, since management of the firewall policies is already covered and physical protection is done by the operational environment.

240 However, we may notice for the running example, that logging might be useful, because experience shows, that there may be reasons to change the defined firewall policy depending on security incidents. Therefore one might want to use two types of logging: To log security incidents (to be defined by unusual amounts of messages of a specific type, for example) and to log, which policy was valid at which time and who made changes to the policy.

241 Note, however, that this functionality is not derived from the objectives listed above, but from general considerations, which belong to a higher level. So the best way to proceed in such case would be to go back to the objectives section or even to the SPD section and to consider, if for example an OSP might be useful, which requires the ability to react to security incidents and to change the firewall policy appropriately. We will not expand on this for our running example, but as a general rule it is recommended to make these kinds of considerations for the main thematic groups as listed above.

242 In general, as is well known from non-security systems development, there will be development cycles in the following sense: During development of a lower level of the PP/ST, one will find ideas and additions for earlier, higher levels. In that case the authors may go back to the higher level and rethink it.

243 A **word of caution**: This guide recommends to use the explanation method, which derives lower levels of the PP/ST from higher levels. Therefore, this cross-check (and other reasons for development cycles in general) should not be used in the sense of the backward method. So it is **not** recommended to artificially add items to the SPD or the list of objectives, only in order to "justify" some additional functionality which "looks good" for the TOE. Only if the cross check reveals that a genuine security problem may have been overlooked and its addition to the SPD can be explained on the SPD level, such addition is recommended.

4.2.4.3 Sub-step 2: Detailed specification of SFRs

244 The following aspect is essential for the specification of SFRs in a PP: The CC doesn't require a PP author to complete all possible operations, he can delegate parts of this to the author of a complying ST. However, he has to give sufficient information to authors of complying STs, that in the end the operations completed in the ST reflect at least the amount of information provided by the security objectives of the PP. In order to achieve this, the PP author has the following options for each operation in an SFR:

- If he sees no restriction for possible completions by the ST author, he can leave the operation completely open;
- He can partly complete the operation, leaving only a restricted choice to the ST author;
- If the conditions, which he wants to impose on the ST author, cannot be modelled by partly completing the operation, he can instead add an application note to the SFR. This application note then defines the restrictions and requirements for the completion of the operation by the ST author. A simple example for a restriction would be that a selection operation must not be completed with the choice of "None".
- He can complete the operation already in the PP.

245 Some of these options will be used in the running example in sub-chapter 4.2.4.6.

246 Note: Application notes can also be used to give other types of information to authors of complying STs or PPs. For example guidance on the choice of additional SFRs may be given or it can be explained how an ST author can transform security objectives for the operational environment into security objectives for the TOE, if the specific TOE has additional security functions covering those objectives. A detailed description of these possibilities is beyond the scope of this guide.

247 For access control policies and information flow control policies we recommend to present them in a condensed form (e.g. as a table) before the SFRs referring to the policy are specified. This is not required by the CC, but has at least three advantages:

- The policy is easier understandable for readers, since it is concentrated in one place.
- It is also easier understandable, because it uses no formal terminology.
- Maintenance of the document is easier, because modifications of the policy, which may occur during development of the PP, are done in exactly one place, not in several SFRs.

248 Note according to work unit APE_REQ.2-3 in the CEM it is required to define all subjects, objects, operations etc., which are used in SFRs in the PP. For the parts described in this guide this is achieved by the proposed tables.

249 **A remark on subjects:**

250 The basic model in CC, Part 2, is based on the assumption, that subjects are entities inside of a TOE operating on behalf of a user. The standard example for this are processes in a multi-user operating system, which are started by users and operate with privileges corresponding to the role of the user. However, the CC recognises, that depending on the circumstances also other entities can be regarded as subjects, as the following passage from CC, Part 2, annex F.5 shows:

"Some information flow control policies may be at a very low level of detail and explicitly describe subjects in terms of processes within an operating system. Other information flow control policies may be at a high level and describe subjects in the generic sense of users or input/output channels."

251 We will demonstrate sub-step 2 for the running example in the sub-chapter 4.2.4.6 and will additionally give some typical examples for often used thematic groups of SFRs later on in the sub-chapter 4.2.4.9.

4.2.4.4 Note on extended functional components

252 The CC allows to define extended functional components in cases, where existing SFRs from Part 2 are not sufficient to describe the functional requirements for a TOE. For details see CC, Part 1, sections 8.3 and C.4, as well as the requirements from CC, Part 3, and the CEM for the families APE_ECD and ASE_ECD.

253 The development of extended components should only be done by experts and guidance on this process is out of scope of this document. However, if a PP author feels the need for the use of extended components, we recommend to first look into published PPs and STs in order to determine if similar extended components have already been defined, and to consider re-using these.

4.2.4.5 Recommended presentation of operations

254 Since the detailed specification of SFRs mainly consists of a careful processing of the operations allowed for SFRs, the visual presentation of the operations is an important step in making the SFRs understandable.

255 For the concept of "operations" and guidance on their use see CC, Part 1, section 8.1 and annex C.2, as well as CC, Part 2, all annexes.

256 The author of a Protection Profile has to explain to the reader how the operations are made visible in the text, in a way that the reader can easily see, how the specific instantiations in the SFRs are derived from the functional components in Part 2 of the CC.

257 We recommend to include the following text at the beginning of the section on SFRs in the PP and to use the form defined in this text:

258 The **refinement** operation is used to add detail to a requirement and thus further restricts a requirement. Refinements of security requirements are denoted in such a

way that added words are in **bold text** and removed words are ~~crossed out~~. If a refinement is added as a separate paragraph to an SFR instead of modifying its wording, this paragraph starts with the word "**Refinement:**" in bold text.

259 The **selection** operation is used to select one or more options provided by the CC in stating a requirement. Selections having been made by the PP author are denoted as underlined text and in addition a footnote will show the original text from CC, Part 2. Selections to be filled in by the ST author appear in square brackets with an indication that a selection is to be made [selection:] and are *italicised*.

260 The **assignment** operation is used to assign a specific value to an unspecified parameter such as the length of a password. Assignments having been made by the PP author are denoted as underlined text and in addition a footnote will show the original text from CC, Part 2. Assignments to be filled in by the ST author appear in square brackets with an indication that an assignment is to be made [assignment:] and are *italicised*. In some cases the assignment made by the PP authors defines a selection to be performed by the ST author, indicated by [selection:] and text, which is is underlined and italicised like *this*.

261 The **iteration** operation is used when a component is repeated with varying operations. The fact, that an iteration operation was used is obvious from the fact, that a component is contained (at least) twice in the PP.
In order to distinguish the individual instances of a component, the component title is amended by showing a slash "/" and an individual name after the component identifier.
Note: For the sake of a better readability this notion may also be applied to some single components (being not repeated) in order to indicate that these SFRs belong to the same functional cluster.

4.2.4.6 Running example: Deriving the SFRs, sub-step 2

262 In sub-step1 we had identified the following thematic groups of SFRS:

-	Identification and authentication	class FIA
-	Information flow control	families FDP_IFC, FDP_IFF
-	Management functions	class FMT

263 In the running example the operations for the information flow SFRs need to be completed to a point, where the basic property of the firewall is reflected (the information flow is regulated between local and remote network and the rules are defined in the firewall policy) and that the management SFRs need to reflect the actions allowed to the operator of the firewall.

For the firewall information flow control policy we need to define the subjects, the (types of) information, the operations, the security attributes and the rules.

264 *Subjects:*

265 In our firewall case, the information flow is between the two network interfaces of the firewall. So these are chosen as subjects.

266 *Information:*

267 This is all network traffic routed through the firewall.

268 *Operations:*

269 The rules simply state that the network traffic is routed according to the traffic policy, which is defined by the operator of the firewall.

270 *TSF Data:*

271 Since the traffic policy is stored in the TOE and the secure operation of the TOE depends on it, it belongs to the "TSF data". Though the CC doesn't require to specify TSF data, it can be useful for easier reference. Therefore we will also include this into the table defining the firewall information flow control policy.

272 With this preparation, the specification of SFRs is quite straightforward. For this example we will specify two thematic groups of SFRs, the group for the information flow control policy and the group for management of the TOE. (Note that in a TOE containing additional other functionality these two groups could also be merged into one group, because they are closely connected.)

273 Note, that according to the explanation method we use as few SFRs as possible to fulfil the objectives. The SFRs are meant to specify the security requirements of a potential user of the TOE, and should not contain implementation aspects (except if there is an OSP prescribing specific implementation details, like use of specific algorithms). As an example, we do not specify the authentication method for the operator, it may be password based, token based, even use a biometric method. Therefore an SFR like FIA_SOS.1 "Verification of secrets" is not used, as it would not apply to a biometric authentication method. Of course the author of a complying ST may add this SFR in the ST, if the specific TOE for this ST uses e.g. password authentication, where this SFR is useful.

274 In the end the corresponding section of the PP will look similar to this:

275 **SFR group for information flow control**

276 This thematic group of SFRs defines the information flow control policy for the TOE, which will be called **Firewall SFP**. For better readability the Firewall SFP is defined in the following table and the SFRs will refer to it:

Type	**Short name**	**Definition**
Subjects	S_LAN	The LAN-interface of the firewall
	S_WAN	The WAN-interface of the firewall
Information	I_Network	All network traffic entering the TOE via S_LAN or S_WAN (Note: Of course a PP author could choose a more specific definition, like IP packets or similar, however we avoid introduction of new technical terms in our running example)
Operations	Block	A network datagram is not sent to one of the interfaces S_LAN, S_WAN
	Route	A network datagram is not blocked (i.e. sent to the interface S_LAN or S_WAN depending on its routing information)
Subject security attributes	-	(No subject security attributes are defined)
Information security attributes	Source address, destination address, source port, destination port, packet type	(Note: These are typical data fields in IP datagrams used for routing - we omit the technical definitions here, these should probably best be given by reference to relevant standards)
Rules	R_Incoming	All network traffic entering the TOE via the WAN-interface will be routed or blocked according to the Traffic_Policy, using the information security attributes.
	R_Outgoing	All network traffic entering the TOE via the LAN-interface will be routed or blocked according to the Traffic_Policy, using the information security attributes.
TSF data	Traffic_Policy	The traffic policy defined by the operator of the TOE.

Table 2: Firewall SFP

277 **FDP_IFC.1 Subset information flow control**

Hierarchical to: No other components.

Dependencies: FDP_IFF.1 Simple security attributes

FDP_IFC.1.1 The TSF shall enforce the Firewall SFP[7] on the subjects, information, and operations as defined in the Firewall SFP (see table 2)[8].

278 **FDP_IFF.1 Simple security attributes**

Hierarchical to: No other components.

Dependencies: FDP_IFC.1 Subset information flow control
FMT_MSA.3 Static attribute initialisation

FDP_IFF.1.1 The TSF shall enforce the Firewall SFP[9] based on the following types of subject and information security attributes: Subjects and information as defined by the Firewall SFP, and for each, the security attributes as defined by the Firewall SFP and [assignment: *list of additional security attributes*][10].

FDP_IFF.1.2 The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold: Rules as defined by the Firewall SFP (see table 2)[11].

FDP_IFF.1.3 The TSF shall enforce **no additional rules**~~the None~~[12].

FDP_IFF.1.4 The TSF shall explicitly authorise an information flow based on the following rules: None[13].

FDP_IFF.1.5 The TSF shall explicitly deny an information flow based on the following rules: None[14].

279 Application note for authors of complying STs: In order to allow a more detailed specification, further security attributes may be added as suitable.

[7] [assignment: *information flow control SFP*]

[8] [assignment: *list of subjects, information, and operations that cause controlled information to flow to and from controlled subjects covered by the SFP*]

[9] [assignment: *information flow control SFP*]

[10] [assignment: *list of subjects and information controlled under the indicated SFP, and for each, the security attributes*]

[11] [assignment: *for each operation, the security attribute-based relationship that must hold between subject and information security attributes*]

[12] [assignment: *additional information flow control SFP rules*], the wording "no additional rules" was added as an editorial refinement instead of "the None".

[13] [assignment: *rules, based on security attributes, that explicitly authorise information flows*]

[14] [assignment: *rules, based on security attributes, that explicitly deny information flows*]

280 Application note: An SFR FMT_MSA.3 is not used here, since the security attributes used in the Firewall SFP are already contained in the network datagrams when entering the TOE, therefore rules for creation of information and default values of security attributes are not applicable.

281 **SFR group for management of the TOE**

282 This thematic group of SFRs specifies the role **operator**, who is the only one allowed to modify the traffic policy of the firewall and who must be authenticated before doing so.

283 **FMT_SMR.1 Security roles**

Hierarchical to: No other components.

Dependencies: FIA_UID.1 Timing of identification

FMT_SMR.1.1 The TSF shall maintain the role~~s~~ operator[15].

FMT_SMR.1.2 The TSF shall be able to associate users with roles.

284 Application note: A role for a "normal" user is not needed here because the interfaces to LAN and WAN do not distinguish any roles and require no identification or authentication.

285 **FMT_SMF.1 Specification of Management Functions**

Hierarchical to: No other components.

Dependencies: No dependencies.

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions: Modification of Traffic_Policy, [assignment: *list of additional management functions to be provided by the TSF*][16].

286 Application note for authors of complying STs: Further management functions may be added as suitable.

[15] [assignment: *the authorised identified roles*]

[16] [assignment: *list of management functions to be provided by the TSF*]

287 **FMT_MTD.1 Management of TSF data**

Hierarchical to: No other components.

Dependencies: FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

FMT_MTD.1.1 The TSF shall restrict the ability to modify, [assignment: *other operations*][17] the Traffic_Policy[18] to the operator[19].

288 Application note for authors of complying STs: In the case, where the firewall has no default traffic policy, and a new policy has to be created, before the firewall can be used, the ST author may want to add the operation "create" to this SFR.

289 **FIA_UID.1 Timing of identification**

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_UID.1.1 The TSF shall allow [assignment: *list of TSF-mediated actions*] on behalf of the **operator**~~user~~ to be performed before the **operator**~~user~~ is identified.

FIA_UID.1.2 The TSF shall require each **operator**~~user~~ to be successfully identified before allowing any other TSF-mediated actions on behalf of that **operator**~~user~~.

290 Application note for authors of complying STs: The ST author may specify actions, which are allowed before identification, however the modification of the traffic policy must not be in this list, since identification and authentication is required for that activity. If the list is empty, FIA_UID.2 shall be used in the ST instead.

291 **FIA_UAU.1 Timing of authentication**

Hierarchical to: No other components.

Dependencies: FIA_UID.1 Timing of identification

FIA_UAU.1.1 The TSF shall allow [assignment: *list of TSF mediated actions*] on behalf of the **operator**~~user~~ to be performed before the **operator**~~user~~ is authenticated.

FIA_UAU.1.2 The TSF shall require each **operator**~~user~~ to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that **operator**~~user~~.

[17] [selection: *change_default, query, modify, delete, clear, [assignment: other operations]*]

[18] [assignment: *list of TSF data*]

[19] [assignment: *the authorised identified roles*]

292 Application note for authors of complying STs: The ST author may specify actions, which are allowed before authentication, however the modification of the traffic policy must not be in this list, since authentication is required for that activity. If the list is empty, FIA_UAU.2 shall be used in the ST instead.

4.2.4.7 Writing the security functional requirements rationale

293 CC Part 1 section A.9.1.2 states: *"The ST [the same holds for a PP] also contains a security requirements rationale, consisting of two sections about SFRs:*

- *a tracing that shows which SFRs address which security objectives for the TOE;*
- *a set of justifications that shows that all security objectives for the TOE are effectively addressed by the SFRs."*

Tracing:

294 The tracing is best addressed by a table, having the SFRs on the rows, and the security objectives for the TOE on the columns, and "X" marking a place in the table if that SFR traces to that security objective. Its purpose is to show that:

a) Each SFR traces to at least one security objective. In other words, each row has to have at least one "X" in it.

b) Each security objective for the TOE has at least one SFR tracing to it. In other word, each column has to have at least one "X" in it.

295 If you have been using the explanation method, the tracing can easily be deduced from the analysis you did to get to the SFRs. The result for the running example is given in section 4.2.4.8.

Justification:

296 The second section of the rationale needs to describe for each security objective for the TOE, how the SFRs meet the objective. If you have done the analysis prescribed by the explanation method, this can be derived from that analysis, as all the elements that are needed are there. They just need to be collected and described in a short, to-the-point, and understandable way. This is best described with our running example, see section 4.2.4.8.

297 Due to the nature of the explanation method, the rationale simply reflects the considerations made when selecting and specifying the SFRs and should therefore be straight forward in most cases.

Functional requirements dependencies rationale:

298 CC, Part 3, states:

"APE_REQ.1.5C Each dependency of the security requirements shall either be satisfied, or the security requirements rationale shall justify the dependency not being satisfied."

299 The recommended way to fulfil this requirement is to produce a table, which lists all SFRs in the first column, the corresponding dependencies as required by CC, Part 2, in the second column, and either the name of the SFRs fulfilling the dependency or a justification for those of the dependencies, which are not satisfied, in the third column.

300 Note: It is important to notice that the fulfilment of dependencies requires the contents of the SFRs to be considered, a mere formal fulfilment is not sufficient. If, for example, two separate information flow policies for different types of information were defined, this would lead to two instantiations of FDP_IFC.1 using the iteration operation, for example FDP_IFC.1 / Firewall SFP and FDP_IFC.1 / VPN Policy. In this case we would also need two instantiations of FDP_IFF.1 to fulfil the dependencies.

301 The consequence of this is: **If a PP uses iterated SFRs, each iteration of an SFR needs its own row in the dependency table.**

302 An example of this is as follows: A TOE provides two cryptographic algorithms, RSA and AES. By iteration this can be modelled using the two SFRs FCS_COP.1 / RSA and FCS_COP.1 / AES. One of the dependencies for the component FCS_COP.1 is [FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]. Its fulfilment has then to be considered for each iteration of FCS_COP.1. For example the dependency could be fulfilled by an instantiation of component FDP_ITC.1, for FCS_COP.1 /AES, if FDP_ITC.1 includes the import of AES keys. At the same time this dependency is fulfilled by FCS_CKM.1 for FCS_COP.1 / RSA if FCS_CKM.1 describes that the TOE generates its own RSA-keys. For better reading one would name these SFRs FDP_ITC.1 / AES and FCS_CKM.1 / RSA, even though they are not iterated in this example.

4.2.4.8 Running example: Writing the security functional requirementsrationale

Tracing:

303 The role of the thematic groups can be seen quite clearly here, since the information flow SFRs contribute to the objectives regarding the traffic routing, while the management SFRs (including the I&A SFRs in this case) address the objective regarding the operator.

	OT.OUTGOING_TRAFFIC	OT.INCOMING_TRAFFIC	OT.OPERATOR
FDP_IFC.1	X	X	
FDP_IFF.1	X	X	
FMT_MTD.1			X
FMT_SMF.1			X
FMT_SMR.1			X
FIA_UID.1			X
FIA_UAU.1			X

Table 3: Tracing of SFRs to objectives for the TOE

Justification:

304 OT.OUTGOING_TRAFFIC and OT.INCOMING_TRAFFIC require the TOE to block all outgoing resp. incoming traffic according to the defined traffic policy. This is exactly, what the rules of the Firewall SFP defined in table 2 state. The SFRs FDP_IFC.1 and FDP_IFF.1 state, that this Firewall SFP is enforced by the TOE, thereby meeting both objectives.

305 OT.OPERATOR requires, that only the operator is able to change the traffic policy for the firewall. The SFR FMT_MTD.1 meets this objective directly by restricting the ability to modify the Traffic_Policy (which is the formal name of the traffic policy as TSF data) to the role operator. The two SFRs FMT_SMR.1 and FMT_SMF.1 support this by defining the roles and management functions needed for FMT_MTD.1. FIA_UID.1 and FIA_UAU.1 require the operator to be authenticated before modifying the traffic policy, which enforces the restriction to the operator effectively.

Functional requirements dependencies rationale:

306 For the running example this table is as follows:

SFR	Required dependencies	Fulfilment of the dependency
FDP_IFC.1	FDP_IFF.1	FDP_IFF.1
FDP_IFF.1	FDP_IFC.1 FMT_MSA.3	FDP_IFC.1 not fulfilled, see below table
FMT_SMR.1	FIA_UID.1	FIA_UID.1 Note: If FIA_UID.2 is chosen instead of FIA_UID.1, this also fulfils the dependency, since it is hierarchical to FIA_UID.1
FMT_SMF.1	None	-
FMT_MTD.1	FMT_SMR.1 FMT_SMF.1	FMT_SMR.1 FMT_SMF.1
FIA_UID.1	None	-
FIA_UAU.1	FIA_UID.1	FIA_UID.1 (Note as above)

Table 4: Dependencies rationale

307 The justification for not including the SFR FMT_MSA.3 is as follows: The security attributes used in the Firewall SFP are already contained in the network datagrams when entering the TOE, therefore rules for creation of information and default values of security attributes for the information are not applicable.

4.2.4.9 Detailed specification of SFRs - Examples for often used thematic groups of SFRs

308 As specific thematic groups of SFRs this guide introduces access control, audit, cryptography, information flow control and security management. These groups were chosen, because they are the most relevant ones for use in practice. The following paragraphs will list an exemplary detailed description of the SFRs used for those groups.

Access control

309 We assume a TOE that enforces a security function policy (SFP) for access control called SFP_AC. The TOE shall regulate the read and write access to a storage server. Furthermore we assume that the stored data consist of data files, which can be read by normal users, but modified only by administrators and of configuration data, which can only be read and modified by administrators. The TOE SFP is consciously chosen generic so it may serve the purpose of a template. For better readability the TOE SFP is defined in the following table and the upcoming SFRs will refer to it:

SFP_AC		
Type	Short name	Definition
Subjects	S_User	User of the TOE (Note: As explained in chapter 4.2.4.3, external entities are allowed as subjects.)
	S_Admin	Administrator of the TOE
Objects	O_Data	Data files
	O_Config	Configuration files
Operations	Read	Read
	Modify	Modify (includes create and delete in this example)
Authentication methods	A_PIN	Authentication with PIN and token (Note: The access control SFRs do not require the listing of authentication methods, however in this example the type of authentication defines a security attribute for the corresponding subject.)
	A_Password	Authentication with password
Security attributes for subjects	(Implicit)	For subjects, the TOE maintains the authentication method used, which implicitly defines the privileges of the subject.
Security attributes for objects:	(None)	In this example the objects are categorised by their type only, without additional attributes.
Rules	R_User	The user can only read data files.
	R_Admin	The administrator can read and modify data and configuration files.

Table 5: Access control SFP SFP_AC

310 In order to provide an access control according to this policy we take several SFRs in account: FIA_UID, FIA_UAU, which together define the requirements for identification and authentication, as well as FDP_ACF and FDP_ACC, which define the attribute based access control policy.

311 **FDP_ACC.2 Complete access control**

Hierarchical to: FDP_ACC.1 Subset access control

Dependencies: FDP_ACF.1 Security based access control

FDP_ACC.2.1 The TSF shall enforce the SFP_AC[20] on all subjects, objects defined by the SFP_AC[21] and all operations among subjects and objects covered by the SFP.

FDP.ACC.2.2 The TSF shall ensure that all operations between any subject controlled by the TSF and any object controlled by the TSF are covered by an access control SFP.

312 **FDP_ACF.1 Security attribute based access control**

Hierarchical to: No other components.

Dependencies: FDP_ACC.1 Subset access control
FMT_MSA.3 Static attribute initialisation

FDP_ACF.1.1 The TSF shall enforce the SFP_AC[22] to objects based on the following: All subjects and objects together with their respective security attributes as defined in SFP_AC[23].

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: Rules for all access methods and access rules defined in SFP_AC[24].

FDP_ACF.1.3 The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: None[25].

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the following additional rules: None[26].

313 Application note: The dependency FMT_MSA.3 will not be fulfilled here, since there is no initialisation of attributes necessary.

[20] [assignment: *access control SFP*]

[21] [assignment: *list of subjects, objects, and operations among subjects and objects covered by the SFP*]

[22] [assignment: *access control SFP*]

[23] [assignment: *list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes*]

[24] [assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]

[25] [assignment: *rules, based on security attributes, that explicitly authorise access of subjects to objects*]

[26] [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*]

314 **FIA_UAU.2 User authentication before any action**

Hierarchical to: FIA_UAU.1 Timing of authentication

Dependencies: FIA_UID.1 Timing of identification

FIA_UAU.2.1 The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

315 **FIA_UID.2 User identification before any action**

Hierarchical to: FIA_UID.1 Timing of identification

Dependencies: No dependencies.

FIA_UID.2.1 The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

316 Application note: In order to define the requirements for the authentication methods in more detail, the PP or ST author may want to add other SFRs like FIA_AFL.1 (which restricts the number of possible failed authentication attempts) or FIA_SOS.1 (in order to require a certain PIN/password quality). In order to explicitly specify the authentication methods mentioned in the SFP (combination of PIN and token, password authentication) and to link the used authentication method with roles, the author may consider usage of components FIA_UAU.5 and FIA_USB.1.

Audit

317 In order to provide a basic audit function, which produces audit logs and allows their review, the following SFRs are chosen: FAU_GEN and FAU_SAR.

318 **FAU_GEN.1 Audit data generation**

Hierarchical to: No other components.

Dependencies: FPT_STM.1 Reliable time stamps

FAU_GEN.1.1 The TSF shall be able to generate an audit record of the following auditable events:

- Start-up and shut-down of the audit functions;
- All auditable events for the <u>not specified</u>[27] level of audit; and
- <u>user login.</u>[28]

[27] [selection, choose one of: *minimum, basic, detailed, not specified*]

[28] [assignment: *other specifically defined auditable events*]

FAU_GEN.1.2 The TSF shall record within each audit record at least the following information:

- Date and time of the event, type of event, subject identify (if applicable), and the outcome (success or failure) of the event; and
- For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, none[29].

319 Application note: The option "not specified" was chosen in FAU_GEN.1.1 for the following reason: According to the explanation method only those events should be chosen for auditing, which are important for the realisation of the security objectives. Therefore it is better to define them individually instead of choosing a predefined set.

320 **FAU_SAR.1 Audit review**

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation

FAU_SAR.1.1 The TSF shall provide the administrator[30] with the capability to read login information[31] from the audit records.

FAU_SAR.1.2 The TSF shall provide the audit records in a manner suitable for the user to interpret the information.

321 **FPT_STM.1 Reliable time stamps**

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_STM.1.1 The TSF shall be able to provide reliable time stamps.

322 Application note: FPT_STM.1 was chosen as a result of a dependency required by FAU_GEN.1. The reason behind this is, that the audit trail entries defined in FAU_GEN.1 require a reliable time as one of their data fields and this time has to be provided by the TOE. Depending on the needs of the specific PP, the author could also have defined an objective for the environment (e.g. for an operating system, if the TOE is a software application) to provide a reliable time function. This would allow to give a rational for not fulfilling the dependency.

[29] [assignment: *other audit relevant information*]

[30] [assignment: *authorised users*]

[31] [assignment: *list of audit information*]

Cryptography

323 This example assumes, that the TOE shall implement two cryptographic algorithms, RSA and AES, according to specific standards. The cryptographic operations are defined by an iterated component FCS_COP.1. In addition, components from FCS_CKM were chosen, since key generation and destruction need to be taken into account.

324 **FCS_CKM.1/RSA Cryptographic key generation**

Hierarchical to: No other components.

Dependencies: [FCS_CKM.2 Cryptographic key distribution, or
FCS_COP.1 Cryptographic operation]
FCS_CKM.4 Cryptographic key destruction

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [assignment: *cryptographic key generation algorithm suitable for RSA*][32] and specified cryptographic key sizes 2048 and 3072 bit[33] that meet the following: [assignment: *list of standards for key generation suitable for RSA as specified in FIPS 186-3*][34].

[32] [assignment: *cryptographic key generation algorithm*]

[33] [assignment: *cryptographic key sizes*]

[34] [assignment: *list of standards*]

325 **FCS_CKM.1/AES Cryptographic key generation**

Hierarchical to: No other components.

Dependencies: [FCS_CKM.2 Cryptographic key distribution,
or FCS_COP.1 Cryptographic operation]
FCS_CKM.4 Cryptographic key destruction

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [assignment: *cryptographic key generation algorithm suitable for AES*][35] and specified cryptographic key sizes 128 and 256 bit[36] that meet the following: [assignment: *list of standards for key generation suitable for AES as specified in FIPS 197 and NIST Special Publication 800-38A*][37].

326 Application note to authors of compliant STs: The choice of a suitable key generation algorithm was left open since it may depend on the implementation of a specific TOE.

327 **FCS_COP.1/RSA Cryptographic operation**

Hierarchical to: No other components.

Dependencies: [FDP_ITC.1 Import of user data without
security attributes, or FDP_ITC.2 Import of
user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1 The TSF shall perform encryption[38] in accordance with a specified cryptographic algorithm RSA[39] and cryptographic key sizes 2048 and 3072 bit[40] that meet the following: FIPS 186-3[41].

[35] [assignment: *cryptographic key generation algorithm*]

[36] [assignment: *cryptographic key sizes*]

[37] [assignment: *list of standards*]

[38] [assignment: *list of cryptographic operations*]

[39] [assignment: *cryptographic algorithm*]

[40] [assignment: *cryptographic key sizes*]

[41] [assignment: *list of standards*]

328 **FCS_COP.1/AES Cryptographic operation**

Hierarchical to: No other components.

Dependencies: [FDP_ITC.1 Import of user data without
security attributes, or FDP_ITC.2 Import of
user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1 The TSF shall perform encryption[42] in accordance with a specified cryptographic algorithm AES with CBC mode of operation and block size 128 bits[43] and cryptographic key sizes 128 and 256 bit[44] that meet the following: FIPS 197[45].

329 **FCS_CKM.4 Cryptographic key destruction**

Hierarchical to: No other components.

Dependencies: [FDP_ITC.1 Import of user data without
security attributes, or FDP_ITC.2 Import of
user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method physical deletion by overwriting the memory data with zeros or the new key[46] that meets the following: None[47].

330 Note, that FCS_CKM.1 were iterated, since keys for RSA and AES have to fulfil different requirements, while FCS_CKM.4 was chosen only once, since all keys are destroyed in the same way.

Information Flow Control

331 For an example in this area we refer to chapter: 4.2.4.6.

Security Management

332 For an example in this area we refer to chapter: 4.2.4.6.

[42] [assignment: *list of cryptographic operations*]
[43] [assignment: *cryptographic algorithm*]
[44] [assignment: *cryptographic key sizes*]
[45] [assignment: *list of standards*]
[46] [assignment: *cryptographic key destruction method*]
[47] [assignment: *list of standards*]

4.2.5 Step 5: Defining the SARs

333 In this section we describe how to choose the security assurance requirements (SARs).

334 CC, Part 1, section A.9.2 states that:

"The SARs are a description of how the TOE is to be evaluated. This description uses a standardised language for two reasons:

- *to provide an exact description of how the TOE is to be evaluated. Using a standardised language assists in creating an exact description and avoids ambiguity.*

- *to allow comparison between two STs. As different ST authors may use different terminology in describing the evaluation, the standardised language enforces using the same terminology and concepts. This allows easy comparison."*

335 While the explanation method describes a systematic and straight forward way how to derive security objectives from the security problem definition and how to derive SFRs from these, the situation is a little bit more complicated for SARs.

336 The reason is as follows: The choice of SARs may often depend on external factors going beyond the discussion of the SPD. Examples of such external factors are:

- A higher assurance level will lead to higher costs of evaluation.

- Laws, regulations or the policy of a customer (or customer group) may require a pre-defined assurance level for certain types of products. This is the case for digital tachograph equipment, digital travel documents or identification documents, for electronic signature equipment and others.

- Other products similar to the TOE were certified for a certain assurance level and marketing requires at least the same level, while cost prohibits to choose a much higher level.

- The developer plans to learn the CC methodology starting with a lower assurance level and plans to use a higher level later on.

- The state of the art of the technology doesn't allow higher assurance levels, because no suitable measures for high security levels were known. This was for example the case for some biometric technology at the time this guide was written, however it was foreseeable that this would change in the future.

- EAL4 is the highest assurance level covered by the mutual recognition agreement. Therefore a developer may chose this level, even if his product would allow a higher one.

337 While the CC requires a rationale for the choice of the chosen SARs, it explicitly allows to take considerations as described above into account. Therefore the statement: "We choose the assurance level EAL4, because this is the level the market requires for products of this type." would be a valid argument in the rationale.

338 The only real restriction for the rationale is that your SARs must not conflict with the rest of the PP: If your TOE must protect against very capable attackers or is placed in a high-risk environment, EAL1 may not be suitable.

339 Having said this, we will briefly present a methodology the PP/ST-author should follow, if

- either he has the freedom to decide on the assurance level without too many external constraints,
- or if he at least wants to know the theoretically suitable assurance level before he takes the external constraints into account.

340 The sub-steps of the methodology are as follows:

341 Sub-step 1:

Use the contents of the security problem definition and possible additional input (e.g. a general security policy of potential customers, the value of assets to be secured by the TOE, potentially a complete risk analysis of the intended environment of the TOE) in order to determine, how resourceful (in terms of expertise, equipment and time) you expect potential attackers to be.

343 Sub-step 2:

Define, as result of the previous sub-step, the ‘level of protection needed’ expressing this level in the language of one of the AVA_VAN components. Appendix B.3.1 of the CEM describes, how a PP/ST-author may do this. For example if you conclude that you want to protect against attackers of "high attack potential", you will choose the AVA_VAN.5 component.

345 Sub-step 3:

Derive the ‘level of assurance wished’ from the ‘protection level needed’ using, amongst others, formal dependencies between the AVA_VAN component chosen and other assurance components. The simplest way to do this, would be to choose the lowest EAL-level containing the component from AVA_VAN identified in the preceding step. As an example: This would imply the choice of EAL6, if you identified AVA_VAN.5 in the preceding step. However, considering additional factors like feasibility of implementation, complexity of evaluation, other resources, etc., you might prefer to choose a lower assurance level, e.g. EAL4, and "augment" it with AVA_VAN.5 (and other components possibly required by dependencies from AVA_VAN.5).

347 As stated above, the PP/ST-author will either use the assurance requirements determined during these stages, or (if he has to take external restrictions into account) he will at least be able to compare the "ideal" and the practical level before final decision.

348 Note that the procedure described above (like the whole "explanation method") is a recommendation. Assurance levels determined by other methods are also allowed by the CC.

4.2.5.1 Running example: Defining the SARs

349 For the running example the sub-steps of the methodology might be as follows:

350 Sub-step 1:

We assume that the firewall is a commercial firewall. Though the assets in the LAN are valuable from the point of view of the owner, usually we will not expect specific directed attacks against the protected LAN. Instead we assume attackers, who want to experiment in the internet (e.g. so-called "script-kiddies") and may use freely available internet scanning tools, which are nevertheless quite powerful against unsecured internet connections.

352 Sub-step 2:

Without detailing the actual numbers here, the results of sub-step 1 may lead to the assumption of attackers with "enhanced basic" potential. The AVA_VAN-level corresponding to "enhanced-basic" attack potential is AVA_VAN.3

354 Sub-step 3:

The first EAL package containing AVA_VAN.3 is EAL4. If we assume for our example that the PP author has no other aspects he wants to consider, he will choose EAL4 as assurance level.

356 The rationale for the choice of assurance components is obvious, it could consist of a slight rewording of the preceding paragraphs.

357 Note: If the author of the PP would think of a firewall for big companies with high valued assets and would assume that very specific attacks might be possible (e.g. industrial espionage), he could come to a completely different result, e.g. that attackers of high attack potential might be relevant. This shows that even if the type of a product and its complete functionality are already defined, the assurance requirements may still differ depending on the considerations described above.

4.2.6 Step 6: Writing the PP introduction

358 According to CC Part 1, section B.4:

359 *"The PP introduction describes the TOE in a narrative way on two levels of abstraction:*

a) the PP reference, which provides identification material for the PP;

b) the TOE overview, which briefly describes the TOE."

360 Most of this section is pretty straightforward and can be derived from CC Part 1 section B.4. We refer to this section, rather than repeating it.

361 The only non-obvious part is the "usage and major security features of the TOE" section of the TOE overview. In our experience the usage is best derived by summarizing the security problem definition, while the major security features are best described by summarizing the security objectives for the TOE. This ensures that the TOE overview is consistent with and fairly complete with respect to the remainder of the PP, without going into overly much detail.

4.2.6.1 Running example: Writing the PP introduction

362 A sample TOE overview for our running firewall example would be:

363 *TOE type:* The TOE type of this PP is a firewall.

364 *Usage and major security features of the TOE:* The firewall is placed between a LAN and a WAN (such as the Internet) to ensure that:

- Entities on the WAN cannot negatively interfere with the LAN;
- Personnel on the LAN can only access parts of the WAN.

365 The firewall does this by allowing an operator to set and maintain a traffic policy for incoming and outgoing network traffic. The firewall will block incoming and outgoing network traffic according to that policy.

366 *Non-TOE hardware/software/firmware:* Apart from network connections with the LAN and the WAN, the TOE requires no additional hardware, firmware or software.

4.2.7 The explanation method: Summary

367 In this chapter we have described two methods for writing a PP:

- The backward method, which is not much work, but not much use either;
- The explanation method which is a lot harder to do, but will provide more insight in the how and why of the PP.

368 We showed that the explanation method consist of the following steps:

- Writing the conformance claims;
- Determining the security problem definition;
- Deriving the security objectives for the TOE and the operational environment including the security objectives rationale;
- Deriving the SFRs including the security requirements rationale;
- Defining the SARs and explain why you have chosen them;
- Writing the PP introduction.

369 And we provided a running example for relevant parts of these steps, showing how they can be derived from each other and what their relationship is.

5 Writing a Security Target

370 Authors, who plan to write PPs and STs need an understanding of the structure and content of PPs and STs that goes into much greater detail than the simple reader's guide in chapter 3. However, in CC v3, this is discussed at significant length in CC Part 1.

371 To prevent duplication or inconsistency, we have not repeated this material in this guide, but in the remainder of this guide we assume that the reader is familiar with the relevant sections and annexes of CC Part 1, and also has some knowledge of CC Part 3 (in particular chapters APE and ASE) and the CEM.

372 It is noted again, that this guide is not meant as a substitution for reading the CC but is intended to give additional guidance based on the contents of the CC.

373 Note also, that we assume that the reader of this section is already familiar with the preceding sections of this guide, in particular with section 3 "Reading Protection Profiles and Security Targets". We will not repeat text from these sections here, which are of course relevant also for writing a Security Target.

374 Security Targets have much in common with Protection Profiles, and therefore many of the rules of the previous chapter hold for this chapter. Rather than repeating everything, we will focus on the differences between the two types of document.

375 In this chapter we will no longer describe the backward method, but concentrate only on the explanation method. Writing STs is almost identical to writing PPs with one obvious difference: As Security Targets have a TOE summary specification, the explanation method has to be extended to cover this. The explanation method for Security Targets consists of the following steps:

- Writing the conformance claims;
- Determining the SPD;
- Deriving the security objectives for the TOE and the operational environment including the security objectives rationale;
- Deriving the SFRs including the security requirements rationale;
- Defining the SARs and explain why you have chosen them;
- Deriving the TOE summary specification;
- Writing the ST introduction.

376 We will revisit each step of this method in the following subsections.

5.1 Step 1: Writing the conformance claims

377 The conformance claims section of an ST is almost identical to that of a PP. The only difference is that since you cannot claim conformance to an ST, you do not have to describe how to claim conformance to that ST (strict or demonstrable).

5.2 Step 2: Determining the security problem definition

378 This section is the same for STs as it is for PPs. As an ST may be for a more specific TOE or for a more specific operational environment there may be more information available than for a PP, but the method is identical to that of PPs.

5.3 Step 3: Deriving security objectives

379 This section is the same for STs as it is for PPs.

5.4 Step 4: Deriving the SFRs

380 The main difference between an ST and a PP is, that all assignment and selection operations need to be completed in an ST. All other considerations are very similar to those for a PP.

381 The ST author may use more specific SFRs than the PP author, as already discussed for the running example in sub-chapter 4.2.4.6, where we discussed the following example: Where a PP only requires a general authentication mechanism, the ST author may already know, that it is password based and may add the SFR FIA_SOS in order to specify quality requirement for the passwords.

382 However, the ST author should keep in mind, that SFRs are meant to specify security requirements from the point of view of the user. SFRs should not be used to describe the technical implementation of the requirements (this would lead to the "backward method", which we recommend to avoid).

5.5 Step 5: Defining the SARs

383 This section is the same for STs as it is for PPs.

5.6 Step 6: Defining the TOE summary specification

384 CC Part 1, section A.10 states that *"The objective for the TOE summary specification is to provide potential consumers of the TOE with a description of how the TOE satisfies all the SFRs. The TOE summary specification should provide the general technical mechanisms that the TOE uses for this purpose. The level of detail of this description should be enough to enable potential consumers to understand the general form and implementation of the TOE."*

385 This section is intended to show the general implementation of the SFRs: If you are wondering whether you are writing down proprietary details, you are probably writing down too much details. A good approach is to describe the implementation of SFRs in conceptual groups, rather than create a dull description of each SFR in isolation. A typical example of a group description would be the following:

386 **Operator identification and authentication** (FIA_UID.1, FIA_UAU.1, FIA_SOS.1 and FIA_AFL.1): The TOE implements these four SFRs with a standard user name/password mechanism, which can only be accessed locally at the terminal. The TOE guarantees that all operator passwords have a length of at least 8 characters, including 1 capital letter, 1 number and one character that is neither a letter nor a number. If an operator fails to input the correct password three times in a row, he will have to wait 10 minutes before he can try another three times.

387 This description could be augmented with one or more screen shots of the mechanism.

388 Note: The CC allows to choose a higher level assurance component, ASE_TSS.2, which requires a more detailed TOE summary specification (TSS). This guide doesn't describe, how to write such kind of TSS, but refers authors, who wish to include this component, to the corresponding passages in the CC (see the definition of component ASE_TSS.2 in CC, Part 3) and the CEM (see the work units for ASE_TSS.2 in the CEM).

5.7 Step 7: Writing the ST introduction

389 Most of this section is pretty straightforward and can be derived from CC Part 1 section A.4. We refer to this section, rather than repeating it.

390 As for PPs, a non-obvious part is the "usage and major security features of the TOE" section of the TOE overview. In our experience the usage is best derived by summarizing the security problem definition, while the major security features are best described by summarizing the security objectives for the TOE. This ensures that the TOE overview is consistent and fairly complete with respect to the remainder of the ST, without going into overly much detail. An example for this was given in chapter 4.2.6.

391 The main difference between an ST introduction and a PP introduction is an additional part, the TOE description, where the physical and logical scope of the TOE have to be described.

392 The physical scope of the TOE is simply a list of all TOE parts. For our firewall this list could be:

- The MinuteGap Platform v1;
- The MinuteGap v18.5 Software Installation CD-ROM;
- The MinuteGap v18.5 Administrator Manual;
- The MinuteGap v18.5 Common Criteria Specific Installation Manual.

393 For the logical scope, one should expand on the list of features already provided in the major security features and provide more detail. One could also borrow from the TOE summary specification.

394 Please refer also to subsections 3.2 “Reading the TOE overview“ and 3.3 “Reading the TOE description“ for additional hints, in particular for the requirement to clearly exclude those security functions, which a reader might expect for a TOE of the given type, but which the actual TOE doesn't support.