Information Technology Security Evaluation Criteria

ITSEC Joint Interpretation Library (ITSEC JIL)

Version 2.0

November 1998

This document is paginated from i to vi and from 1 to 65

Table of contents

0 Introduction

1 In order for this document to address the direct problems associated with the everyday use of the [ITSEC] the four countries originally involved in its production (France, Germany, the Netherlands and United Kingdom) formed a Joint Interpretation Working Group (JIWG).

2 This document contains agreed interpretations of the [ITSEC] which have been derived from observations raised by the four European countries through their application of these criteria.

3 In applying the [ITSEC] over numerous evaluations considerable experience has been gained within these countries. The application of these criteria has given rise to a number of observations, many of which are common, within these european countries. These observations required the national schemes to make interpretations to their ITSEFs to allow evaluations to continue.

4 The number of observations raised over the years by the national schemes has now made it necessary to discuss these interpretations and reach a common agreement. Such an agreement is needed to promote their rationalisation into a common document, entitled the ITSEC Joint Interpretation Library (ITSEC JIL) with a view to supporting mutual recognition.

5 The problems that were observed within each national scheme during the course of live evaluations, resulted in national interpretations. These documents were categorised into topics. Each of these topics then formed a chapter in the ITSEC JIL.

0.1 Scope

6 This document contains an agreed set of interpretations to promote mutual recognition. It is intended to supplement the [ITSEC]. This document is not in itself an amendment to the [ITSEC].

7 The manner in which this document is adopted in each country is not within the scope of this document.

8 However, a common thread in producing the interpretations in this document was the effect of the [ITSEM]. In many cases [ITSEM] was able to provide enough information to allow a consistent interpretation to be made.

0.2 Terminology

9 This document uses two main terms for each topic, these are background and interpretation.

0.2.1 Background

10 The background is intended to provide an introduction into the topic and to focus on the main problem. References out to the appropriate parts of the [ITSEC] and [ITSEM] are often provided along with any relevant discussion.

0.2.2 Interpretation

11 Interpretations are provided to allow a common understanding of the original [ITSEC] concepts in order to facilitate the evaluation process.

0.3 Process Maintenance

12 This document will be maintained by the JIWG who will regularly review future national interpretations which are submitted.

1 Glossary

1.1 Vulnerability and Critical Mechanism

1.1.1 Background

13 The term “security weakness” is not defined within the [ITSEC] glossary. However, it is used within the text of [ITSEC] and twice within the glossary to define vulnerability and critical mechanism.

14 [ITSEC 3.5] All critical security mechanisms (i.e. those mechanisms whose failure would create a security weakness) are assessed for their ability to withstand direct attack.

15 [ITSEC 6.22] **Critical Mechanism:** a mechanism within a TOE whose failure would create a security weakness.

16 [ITSEC 6.76] **Vulnerability**: a security weakness in a TOE (for example, due to failure in analysis, design, implementation or operation).

17 The only occurrence of “security weakness” except for the glossary is in [ITSEC 3.5]. Vulnerability is, however, defined within the glossary [ITSEC 6.76] and used to define critical mechanism [ITSEC 6.22] which leads to a circular reference and an undefined term, “security weakness”. Therefore, an interpretation is provided of the present definition of “vulnerability” as defined in [ITSEC 6.76].

1.1.2 Interpretation

18 For consistency the definition of “vulnerability” and “critical mechanism” within the [ITSEC] will be interpreted as follows:

19 **Vulnerability:** a weakness in the construction or operation of the TOE that could prevent it from meeting one or more of its security objectives (for example, possibility of deactivating, bypassing, corrupting, circumventing or directly attacking security enforcing functions and mechanisms [ITSEC 3.21]).

20 **Critical Mechanism:** a security enforcing mechanism within the TOE which is susceptible to direct attack (see section 6.3).

1.2 End-user

1.2.1 Background

21 [ITSEC 6.33] defines an end-user as a person in contact with a TOE who makes use only of its operational capability. This definition does not explicitly take into account the fact that the TOE may not directly be accessed by a person but by other

external components instead. This leads to the following interpretation of the term end-user.

1.2.2 Interpretation

22 A person or an active entity in contact with a TOE that makes use only of its operational capability.

2 Rigour

2.1 Background

23 The [ITSEC] stipulates:

- in section 0.12 that "the verbs state, describe and explain are used within criteria to require the provision of evidence of increasing levels of rigour. *State* means that relevant facts must be provided; *describe* means that the facts must be provided and their relevant characteristics enumerated; *explain* means that the facts must be provided, their relevant characteristics enumerated and justifications given."

- and in section 4.14 that "There is a general need for greater rigour and depth in the evidence provided at higher evaluation levels. This is reflected in the progressive use of the verbs *state*, *describe* and *explain* at different levels in many criteria for content and presentation which do not otherwise change."

24 An evaluation is based on the evaluation deliverables provided by the sponsor/ developer. These deliverables provide information through which the evaluators are able to understand the TOE and evidence by which the evaluators gain sufficient confidence that the TOE meets the evaluation criteria.

25 The accepted principles of [ITSEC] link assurance levels (which characterise the rigour of the evaluation results) with the production of evidence which in itself becomes more detailed and formal as the evaluation level increases. This principle is demonstrated in the description of the TOE, which is itself refined for the description of the architectural design/detailed design and implementation. Such descriptions are required, as the evaluation level increases, to show an increase in the level of rigour through the use of the verbs state/describe/explain.

26 The specified level of rigour provided by the sponsor/developer in the documentary evidence is firmly aligned in the [ITSEC] criteria to the evaluation levels. At the highest level of rigour "explain" means that the evidence must be complete and the solution justified in a culture shared by both developers and evaluators (meaning of the terms, everyday notions, logic of the predicates, set theory).

27 At the lowest level of rigour, the verb "state" means that the evidence provided by the sponsor/developer need only be specified to the most abstract form in respect of the properties of non-ambiguity, consistency and completeness. For example a procedure (delivery, generation, start-up) may be specified by a set of abstract actions in the form of a list (of functions, tests, mechanisms). Such a list is required to be exhaustive in stating all elements but does not need to contain any other information.

28 The limits of rigour involving "state" and "explain" have been set in the previous paragraphs. The verb "describe" lies between the two meanings in that the evidence must be complete building on that which would be provided for "state" in the most

abstract form in respect of the properties of non-ambiguity, consistency and completeness, excluding any evidence that would provide justification. The evidence must be provided at a level of detail that can be used as a basis for the implementation of the concept.

2.2 Interpretation

29 The verbs “state”, “describe” and “explain” are used to ensure that precise information is provided for the appropriate level of rigour:

30 “State” means that all relevant facts are listed.

31 “Describe” means that all relevant facts are listed **and their characteristics are fully detailed. Information provided on the solution must be refined to a level of detail that can be used as a basis for the implementation of the concept**.

32 “Explain” means that all relevant facts are listed and their characteristics are fully detailed. Information provided on the solution must be refined to a level of detail that can be used as a basis for the implementation of the concept. **In addition, a complete justification of the solution must be given**.

3 ITSEC Figure 4

3.1 Background

33 [ITSEC 3.4] introduces [ITSEC figure 4]: "As a minimum, the sponsor's vulnerability analysis must consider all the information specified in figure 4 for the evaluation level in question" (i.e. a search for vulnerabilities is to be performed using part of the total information provided by the sponsor for the evaluation level).

34 It is not clear that the information contained in the detailed design must be studied when carrying out the effectiveness analysis. It is also unclear for which aspects of effectiveness analysis the detailed design is to be used at the E2 level.

35 The information listed in [ITSEC figure 4], which depends on the selected evaluation level, represents a minimum requirement for the information to be used. This statement is based on [ITSEC 3.4] which reads that the search for vulnerabilities is to be performed using part of the total information provided by the sponsor.

36 However, the following reasons can be provided for considering the detailed design in the effectiveness analysis:

a) the detailed design is available for E2 [ITSEC E2.8];

b) the mapping of the security functions to the mechanisms must be shown in the detailed design and the mechanisms must be specified; and

c) the [ITSEC E2.8] explicitly requires that this specification (of the mechanisms) must be sufficiently detailed to allow an analysis of the relationship between the mechanisms (binding analysis) to be made.

3.2 Interpretation

37 The title of [ITSEC figure 4] is interpreted as "Information used in Effectiveness Analysis". Depending on the TOE, the sponsor may have to use additional documents for the effectiveness analysis.

38 [ITSEC figure 4] describes the documentation to be used as a minimum for producing the sponsor's effectiveness analysis. However, the evaluators shall use all documentation provided by the sponsor and the correctness analysis results.

4 Security Target

4.1 TOE Description

4.1.1 Background

39 In [ITSEC] there is currently no explicit requirement to precisely define the scope and boundary of the TOE. This might lead to problems during the course of an evaluation. It is agreed that the exact scope and boundary of the TOE should be defined in the security target.

40 In order to achieve reproducibility and repeatability of evaluations, it is necessary to be able to identify precisely the system or product that undergoes evaluation. It is recognised that for a concurrent evaluation the precise information required may not be initially available. However, by the end of the evaluation the security target must be complete. The TOE should thus be uniquely identified in the Evaluation Technical Report (ETR) and also in the certification report, so that the reader knows exactly what has been evaluated.

4.1.2 Interpretation

41 [ITSEC 2.4] must be interpreted as including information on the precise scope and boundary of the TOE, so that no ambiguities can exist between what is and what is not part of the evaluation.

42 The scope and boundary of the TOE must be described both in a physical way (i.e. by listing the hardware and/or software components/modules) and a logical way (i.e. by listing the functionalities offered by the TOE). All dependencies on external hardware or software components outside the TOE must be documented to the appropriate level of rigour. Where the TOE is a product the evaluation shall include any interface, but not the external component behind that interface.

4.2 Security Target Description

4.2.1 Background

43 In [ITSEC 2.4-2.26] a complete description is given about what should be included in a security target. In [ITSEC En.2] only part of this description is repeated. [ITSEC 2.4-2.26] applies to all E-levels.

4.2.2 Interpretation

44 In [ITSEC En.2], the sentence:

> "In the case of a system, in addition the security target shall include a System Security Policy (SSP) identifying the security objectives and the threats to the system. In the case of a product, in addition the security target

> shall include a rationale, identifying the method of use for the product, the intentioned environment and the assumed threats within the environment."

must be interpreted as follows:

> "The security target shall fulfil the requirements described in the "security target" section of Chapter 2 (paragraphs 2.4-2.26), making sure the requirements are those appropriate for the evaluation level and the type of TOE concerned (product or system)".

4.3 Threats and Security Objectives

4.3.1 Background

45 [ITSEC] presents an unclear view on the relationship between security objectives and threats. In particular:

- [ITSEC 1.25-26], [ITSEC Figure 3] and [ITSEC 2.11] clearly indicate that the security objectives are derived from the identified or assumed threats.

- [ITSEC 6.63] contradicts this view by stating (as part of the definition of a security target) "It will also specify the security objectives, the threats to those objectives..."; this indicates that the security objectives are identified first. [ITSEC 1.11] is also supportive of this view.

46 Clarification has been sought on the order of specifying the threats and security objectives. There is no specified order within the text of [ITSEC]. This undefined hierarchy has led in some circumstances to the circular specification of threats and security objectives within a security target. This circular referencing has led to a failure to meet the criteria for the appropriate level of rigour when matching a threat to an security objective or vice versa.

4.3.2 Interpretation

47 There are no criteria requirements on the sponsor as to the order of identifying either the threats or security objectives in the SSP within the security target. However, whichever is first specified, the latter must be specified, at the appropriate level of rigour, to a lower level of refinement to avoid the circular specification of a threat and a security objective identified to counter it.

5 Detailed Design

5.1 Basic Component

5.1.1 Background

48 In [ITSEC 6.10] basic component is defined as “a component that is identifiable at the lowest hierachical level of specification produced during detailed design”.

49 [ITSEC E2.15] includes the requirements “...The configuration list provided shall enumerate all basic components out of which the TOE is built...” and “...The configuration control system shall ensure that the TOE under evaluation matches the documentation provided and that only authorised changes are possible.”

50 The requirements in [ITSEC E2.15] can lead to two problems:

a) in general, the basic components in a configuration control system do not necessarily match those identifiable at the lowest hierarchical level of specification produced during detailed design; and

b) at E2, an evaluator does not have access to the source code and hardware drawings. How therefore, can the evaluator check that “the TOE under evaluation matches the documentation provided”?

5.1.2 Interpretation

51 At E2 and higher, the TOE documentation, including the detailed design, must be maintained under configuration control and be representative of the TOE under evaluation. Such documentation must be consistent with the TOE.

52 At E3 and higher, the basic components of the detailed design may be different from the basic components defined for configuration control purposes. However, traceability of the basic components at detailed design level to the basic components for configuration control must be provided [ITSEC En.11, n>2].

5.2 Realisation

5.2.1 Background

53 The exact meaning of the word “realisation” in the terms of its use within [ITSEC] is not defined in the [ITSEC] glossary. The only occurrence of “realisation” in [ITSEC] is in [ITSEC En.8, n>1].

54 [ITSEC En.8, n>1] The detailed design shall state/describe/explain the realisation of all security enforcing and security relevant functions.

5.2.2 Interpretation

55 In [ITSEC En.8,n>1], realisation is interpreted as the result of the refinement from one level of representation to the next.

6 Mechanism

6.1 Nature of Mechanism

6.1.1 Background

56 The subject of mechanisms, as used in [ITSEC], has not been consistently well understood within the evaluation community. This topic attempts to promulgate a clearer understanding of the nature of mechanisms. Interpretations are provided on the distinction between security functions and mechanisms, and the distinction between mechanisms and components, and guidance is given on the use of security functions and mechanisms.

- **Security mechanism:** is defined in [ITSEC 6.59] as "the logic or algorithm that implements a particular security enforcing or security relevant function in hardware and software."

- **Component:** is defined in [ITSEC 6.14] as "an identifiable and self-contained portion of a Target of Evaluation. "

6.1.2 Interpretation

57 The following interpretations are provided:

- **Security Functions and Mechanisms:** Security functions specify *what* security functionality is required. Security mechanisms specify *how* it is to be provided. Security mechanisms give an abstract realisation of Security functions.

- **Mechanisms and Components:** Security mechanisms specify *how* security functionality is to be provided. Components specify *where* it is to be provided. Thus, whilst mechanisms give a logical model of the TOE, components provide a basis for coding and production of operational documentation detail.

6.1.3 Guidance

58 The following guidance is provided on the use of security functions and mechanisms.

59 Within correctness, mechanisms occur predominantly in the Detailed Design and Implementation work packages.

60 [ITSEC En 8.9, n>1] the Detailed Design is required to identify and specify security mechanisms and specify how the security mechanisms provide the security enforcing functions. The identification and specification of security mechanisms within the Detailed Design contributes to the traceability of security functions and

provides a basis for the employment of mechanisms within the effectiveness analyses.

61 [ITSEC En.13, n>2] requires (after requirements for correspondence in [ITSEC En.12]) that tests cover not only all security functions but also all security mechanisms. This dual requirement assumes particular significance where there is not a one to one mapping between functions and mechanisms; e.g. if a security enforcing function is implemented by a pair of mechanisms then it is necessary for both mechanisms to be tested.

62 [ITSEC] mentions mechanisms in the context of all effectiveness work packages except the Operational Vulnerability Assessment. Within Suitability, Binding, Construction Vulnerability Assessment and Ease of Use [ITSEC] mentions mechanisms in conjunction with security enforcing functions (mechanisms providing the abstract implementation of security enforcing functions). However, in cases where there is not a one to one mapping of a security enforcing function to a mechanism, the criteria should be applied to both security enforcing functions and mechanisms. In extension of [ITSEC 3.21 to 3.25] the Strength of Mechanism is concerned with all mechanisms susceptible to direct attack (see section 6.3)

63 As it may be possible to have a security function implemented by one or more security mechanisms, it is necessary to consider the binding of one or more of these mechanisms and the possibility of deactivating at least one of them.

6.2 Mechanism Types

6.2.1 Background

64 [ITSEC] introduces the following concepts:

- **Security Mechanism:** this term only occurs under correctness aspects, and is defined in [ITSEC 6.59] as “the logic or algorithm that implements a particular security enforcing or security relevant function in hardware and software”.

- **Security Enforcing**: it is defined in [ITSEC 6.58] as “that which directly contributes to satisfying the security objectives of the TOE”.

- **Critical Mechanism**: as defined in [ITSEC 6.22] it is “a mechanism within a TOE whose failure would create a security weakness”.

6.2.2 Interpretation

65 The following interpretations are provided:

- **Security Enforcing Mechanisms**: this is a subset of security mechanisms. It consists of those security mechanisms which directly contribute to satisfying the security objectives of the TOE.

- **Critical Mechanism**: this is interpreted in section 1.1.2 of this document.

- **Supporting Protection Mechanism**: this is a mechanism used by the TOE which contributes to the security of the TOE but is not part of the TOE [ITSEC 4.20, En.5 n>1].

66 These relationships are expressed in the following diagram:

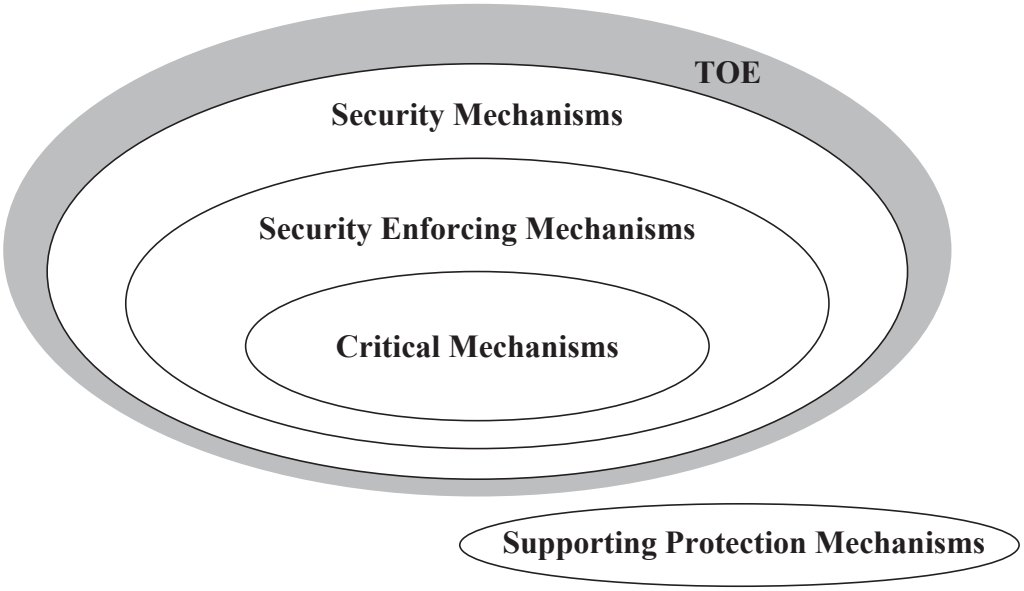

Fig. 6.1 -Relationships between types of mechanisms

67 In accordance with this interpretation, it is possible to have a TOE without critical mechanisms. In this case, the strength of mechanisms analysis is limited to a rationale (at the appropriate level of rigour) which justifies why security enforcing mechanisms are not considered to be critical. However, the claimed minimum strength of mechanisms is also used in the vulnerability analysis according to [ITSEM 3.4.20], "The search for exploitable vulnerabilities is limited by the amount of information provided according to the evaluation level and the level of expertise, opportunity, and resources corresponding to the claimed minimum strength of mechanisms" (see section 6.4 of this document).

6.3 Direct Attacks

6.3.1 Background

68 The construction correctness analyses check the correct realisation of individual security enforcing functions from their specification in the security target to their implementation in code or hardware. The construction effectiveness analyses first check the adequacy of the security enforcing functions to counter the threats, and the ability of the combination of the security enforcing functions and their

realisation to provide a secure, integrated whole. This still leaves the possibility that a weakness might exist in the form of a mechanism which, because of the nature of its conception, can be overcome by direct attack. The strength of mechanisms analysis is required to identify such mechanisms and assess their ability to withstand direct attack. Note that the strength of mechanisms analysis thus assesses the conception of mechanisms conceived to provide particular security functionality, rather than considering vulnerabilities which may result from the inadequacy, poor combination or incorrect realisation of security enforcing functions.

69 The claim for a minimum strength of mechanisms is seen to complement the specified assurance level, by making a claim in respect of those mechanisms which, by nature of their essential conception, each contain a weakness which leaves them vulnerable to direct attack.

70 According to [ITSEC 3.21 to 3.25] the strength of mechanisms analysis applies to security enforcing mechanisms only. The following discussion therefore first confines its attention to security enforcing mechanisms. However the role of security relevant mechanisms in the context of the minimum strength of mechanisms claim is then considered.

71 In general an attack is an attempt to violate the TOE's security objectives by exploiting a weakness of one of the following types:

a) a flaw in traceability and implementation (i.e. where the TOE is not correct),

b) the inability of countermeasure(s) to adequately counter a threat (i.e. where the TOE is not suitable),

c) failure of the TOE's security enforcing functions and mechanisms to provide an integrated and effective whole (i.e. where the TOE is not effectively bound),

d) a mechanism which, because of the essential nature of its conception, possesses a residual weakness in its underlying algorithm, principles or properties,

e) insecure operation which is not easily detectable by the authorised user (i.e. where the TOE's ease of use is not effective).

72 Direct attack typically involves manipulation of inputs to and/or outputs from the mechanism which are within its specification.

6.3.2 Interpretation

73 Therefore by inference, a direct attack is understood to be an attempt to violate the TOE's security objectives by exploiting a weakness in the underlying algorithm, principles or properties of a particular mechanism.

74 Direct attacks can be carried out on critical mechanisms and on specific security mechanisms not security enforcing but security relevant. The strength of mechanisms analysis is required for critical mechanisms. However the same analysis is also appropriate for those security relevant mechanisms that are susceptible to direct attacks.

75 As part of evaluator actions required by [ITSEC 3.24], the evaluator shall check that:

- all attacks relevant to the individual TOE are listed taking into account the different types of weaknesses outlined in section 6.3.1. Direct attacks are to be identified;
- all critical mechanisms are identified;
- all security relevant mechanisms that are susceptible to direct attack are identified;
- a rationale is provided why other mechanisms within a TOE are not susceptible to the identified direct attack and thus need not to be analysed under the strength of mechanisms analysis and
- all direct attacks are analysed under the consideration of the strength of mechanisms claim even those on security relevant mechanisms that are susceptible to direct attacks.

6.4 Strength of Mechanisms

6.4.1 Background

76 According to [ITSEC 2.25], “Every security target shall specify a claimed rating of the minimum strength of the security mechanisms of the TOE against direct attack”. The question is whether this also applies to a TOE that does not contain any critical mechanisms.

77 At the requirements stage, there is not sufficient evidence provided to determine, whether the TOE contains any critical mechanisms or not. The relevant evidence is provided in the strength of mechanisms analysis.

78 According to [ITSEM 3.4.20], “The search for exploitable vulnerabilities is limited by the amount of information provided according to the evaluation level and the level of expertise, opportunity, and resources corresponding to the claimed minimum strength of mechanisms.”

79 In [ITSEM 6.C.30 b)] no allowance is made to include study time in mechanisms analysis. There may be instances where the application of study time is questionable.

6.4.2 Interpretation

80 Given the above, a minimum strength of mechanisms claim shall be provided in the security target for all TOEs even if there are no critical mechanisms.

81 Should the TOE contain critical mechanisms, the rating for strength of mechanisms shall be recorded in the certificate and certification report.

82 Should the evaluation determine that the TOE does not contain any critical mechanisms, this shall be stated in the certificate. The claim for strength of mechanisms shall be stated as a minimum in the certification report in the context of effectiveness to indicate that the vulnerability analysis has been performed accordingly.

83 The minimum strength of mechanisms claim also provides a scale which shall be used to determine whether or not vulnerabilities in the TOE generally are exploitable in practice.This means, as a minimum, the examination of known and potential vulnerabilities is to be performed according to the level of expertise, opportunity and resources corresponding to the claimed minimum strength of mechanisms. Determination of whether or not a vulnerability is exploitable in the TOE's environment involve consideration of the levels of expertise, opportunity and resources required for its exploitation.

84 There are instances where study time could be a factor and should be considered in the strength of mechanisms analysis.

6.5 Strength of Mechanisms at E1

6.5.1 Background

85 At E1, there is no requirement to provide detailed design documentation. Since the specification of mechanisms is part of detailed design, the evaluator does not have the necessary information to evaluate the strength of mechanisms claim.

6.5.2 Interpretation

86 As stated in [ITSEC 3.22], the sponsor must provide evidence to support a strength of mechanisms assessment. Such evidence may require more documentation than the requirement given in [ITSEC E1.1], in which case a description of the detailed design of those mechanisms will be required.

7 Source Code

7.1 Background

87 This chapter interprets the [ITSEC] in terms of what are the evaluation requirements for the analysis of source code? In the same context [ITSEC] also refers to hardware drawings. While hardware drawings, are not considered within this chapter, it is likely that the same approach could be taken as that applied to source code.

88 At E3, no explicit effectiveness analysis of source code and hardware drawings is required. This appears to be an inconsistency in the correspondence between E3 level and B1 TCSEC class; the B1 class requires a systematic analysis of vulnerability at the source code level. However, it is acknowledged that the mapping between [ITSEC] and [TCSEC] given in [ITSEC 1.39] is only one of general correspondence.

7.2 Interpretation

89 At E3 and above, source code is a required [ITSEC] deliverable for evaluations. At E4 and above, source code must be considered in effectiveness analyses.

90 Source code shall be used in the following parts of evaluation:

a) Correctness analysis (at E3 and above) where the source code is checked in traceability and test coverage analysis;

b) Effectiveness analysis (at E4 and above) according to the [ITSEC figure 4].

7.2.1 Correctness Analysis of Source Code

91 There is no requirement for detailed examination of the source code as a feature of correctness evaluation. Source code is provided to check the traceability of functions to their physical representation.

92 At E3 and above, the availability of source code provides increased scope for devising penetration tests and additional functional tests. This does not involve evaluating the source code as such, but using relevant parts of it as a means of understanding how best to carry out such tests.

93 According to the hierarchical concept of assurance in [ITSEC] there is a general need for greater rigour and depth in the evidence provided at higher evaluation levels [ITSEC 4.14, 4.22]. At E5 and E6, this principle is reflected at the implementation level in requirements for additional characteristics. These characteristics are the complete structuring of the source code into small, comprehensible and separate sections [ITSEC En.11, n>4], and the exclusion of unnecessary functionality from security relevant and security enforcing

components according to [ITSEC En.8, n>4]. Since the absence of functionality cannot be checked by functional testing, this could be supplemented by source code analysis.

94 E5 and E6 also permit a more detailed analysis, such as determining branching conditions, which allows a higher level of assurance to be achieved.

95 At E3 and above, source code provides the basis for test coverage analysis. [ITSEM 4.5.72 a)] suggests that the E3 requirements are met if every statement of security enforcing source code is tested at E3. The determination of which source code components are security enforcing is provided by the traceability, which is to basic components level for E3 and E4 and to the functional unit level [ITSEC 6.38] for E5 and E6. It may, nonetheless, be possible to identify specific statements within a source code component which are security enforcing (with the remainder categorised as security relevant) and which therefore need to be covered during testing. For security enforcing components, where this identification is not possible, all source code lines need to be covered during testing.

7.2.2 Effectiveness Analysis of Source Code

96 [ITSEC figure 4] relates the information to be used to perform the effectiveness analysis to the evaluation level. Hence, the general principle of increasing rigour and depth in [ITSEC 4.14] ([ITSEM 4.5.14]) also applies to effectiveness. Although the [ITSEC] does not refer to the source code explicitly under effectiveness requirements, at E4 and above the source code has to be considered, and with increasing depth at higher levels.

97 Effectiveness analysis of source code concerns specifically the binding analysis, vulnerability analysis and the specification of penetration tests. [ITSEM 4.5.35] provides generic methods using source code to analyse potential vulnerabilities and to specify penetration tests.

98 Effectiveness analysis of source code could include, for example, the analysis of:

a) procedure calls;

b) global and local variables;

c) pointers;

d) indirect interaction mechanisms such as signals, semaphore, shared memory etc.

99 Binding analysis, vulnerability analysis and the specification of penetration tests from source code may also include covert channel analysis. For the other effectiveness aspects, the main use of source code would be to clarify the precise behaviour of the TOE but systematic analysis of source code would not be required.

8 Development Environment

8.1 Site Visits

8.1.1 Background

100 [ITSEC En.17, En.23, En.34, n>1] require that the documented procedures must be checked. This concerns the configuration control aspect, developers' security aspect and delivery procedures.

101 The evaluator must check the development environment procedures; at E2 and higher, this requires a check as to whether the developer of the TOE applied the documented procedures.

8.1.2 Interpretation

102 [ITSEC En.17, En.23, En.34, n>1] require that the evaluator checks that the "documented procedures are applied". To fulfil this requirement, the evaluator must carry out one or more development and production site inspections.

103 The objectives of these [ITSEC] requirements on the development and production environment (configuration control and developer's security) are to ensure the integrity of the TOE as well as the confidentiality of the development documentation.

104 In order to guarantee the integrity of the delivered TOE [ITSEC En.32, n>1], the evaluation of these procedures must include the production and delivery processes [ITSEC 4.23, 6.50].

105 [ITSEC En.22, n>1] stipulates that "The information on the security of the development environment shall state/describe/explain how the integrity of the TOE and the confidentiality of the associated documentation are maintained". If the evaluation raises serious concerns over the integrity of the TOE or confidentiality of the development documentation and problem reports or corrective measures have been ineffective, the TOE must be rated E0. If confidentiality is not an issue for the TOE, then the sponsor shall provide justification.

106 The first site visit has to be scheduled as early as possible. In the case of a concurrent evaluation, this will allow corrective action to be taken, if necessary. It may be necessary during concurrent evaluations to perform more than one site visit to the same site to allow the checking of all development phases. In the case of a consecutive evaluation, an early site visit permits termination of the evaluation if serious deficiencies in the applied procedures emerge. This avoids unnecessary evaluation effort.

107 Documented procedures to ensure the integrity of the TOE and the confidentiality of the associated documents are not linked to the evaluation level. The specification

of detailed requirements concerning these aspects is the responsibility of each national scheme.

8.2 Developer's Quality Management Procedures

8.2.1 Background

108 [ITSEC En.16, n>1] states "The information on the configuration control system shall state/describe/explain how it is used in practice and applied in the manufacturing process in accordance with the developer's quality management procedures."

109 In accordance with chapter 16 of this document, manufacturing process in [ITSEC En.16, n>1] is interpreted as development and production process.

8.2.2 Interpretation

110 The above criteria do not require that additional quality management procedures to those required by [ITSEC] are applied to the development and production process. However, the evaluator must check that the configuration control system is correctly applied in accordance with all relevant quality management procedures. The developer shall provide these relevant procedures.

9 Configuration Control

9.1 Configuration List

9.1.1 Background

111 At E1, [ITSEC E1.15-E1.17] require the evaluator to check that the configuration list states how and where the TOE is uniquely identified.

112 [ITSEC En.15, n>1] requires that “the configuration list provided shall enumerate all basic components out of which the TOE is built”. It is not clear if the basic components are the only elements which the configuration control system has to manage.

9.1.2 Interpretation

113 At E1, the evaluators must check the configuration list against the evaluated TOE. This should be understood to involve checking that the evaluated TOE is that which the configuration list assets it to be.

114 [ITSEC En.15, n>1] states “The configuration control system shall ensure that the TOE under evaluation matches the documentation provided and that only authorised changes are possible”. To achieve this objective, it is therefore necessary to include in the configuration list anything which can cause a change in the TOE.

115 In [ITSEC En.15, n>1], the sentence “the configuration list provided shall enumerate all basic components out of which the TOE is built”, is interpreted as “the configuration list provided shall enumerate all basic components out of which the TOE is built as well as all the elements necessary for building and testing the TOE”.

10 Delivery

10.1 Background

116 [ITSEC 4.31] "Delivery and Configuration" covers the requirements for procedures to be in place to maintain the security of the TOE or its components during initial delivery and that of any subsequent modification delivered to the user. Such procedures are to ensure that the security protection offered by the TOE is not compromised during delivery from the production site to the installation site. This includes intermediate stages.

117 [ITSEC En.32, n>1] "A procedure approved by the national certification body for this evaluation level shall be followed". This procedure must guarantee the authenticity of the delivered TOE.

10.2 Interpretation

118 This interpretation sets objectives for each level of evaluation:

119 E1 **The delivery procedure shall be documented**.

120 E2 The delivery procedure shall be documented **and applied. A method shall exist for the receiver to detect obvious modification which has been made during delivery.**

121 E3 The delivery procedure shall be documented and applied. A method shall exist for the receiver to detect obvious modification which has been made during delivery. **It shall be possible to detect that an unauthorised agent has initiated the delivery.**

122 E4 The delivery procedure shall be documented and applied. A method shall exist for the receiver **to detect any modification to the TOE** which has been made during delivery. It shall be possible to detect that an unauthorised agent has initiated the delivery.

123 E5 The delivery procedure shall be documented and applied. A method shall exist for the receiver to detect any modification to the TOE which has been made during delivery. It shall be possible to detect that an unauthorised agent has initiated the deliver. **The delivery path shall be trusted (trusted transfer[1]).**

124 E6 The delivery procedure shall be documented and applied. A method shall exist for the receiver to detect any modification to the TOE which has been made during delivery. It shall be possible to detect that an unauthorised

1. The defined path of transfer whose integrity is trusted at all stages by the originator and the receiver. The use of this transfer path can only be initiated by the originator and the receiver.

agent has initiated the delivery. The delivery path shall be trusted (trusted transfer[1]). **The originator and the receiver shall use a method of authentication.**

1. The defined path of transfer whose integrity is trusted at all stages by the originator and the receiver. The use of this transfer path can only be initiated by the originator and the receiver.

11 Sampling

11.1 Background

125 Sampling is a defined procedure whereby some part of an evaluation deliverable is examined and assumed to be representative of the entire evaluation deliverable.

126 [ITSEC] identifies two evaluator actions where sampling is explicitly acceptable:

a) "Use the library of test programs to check by sampling the results of tests" [ITSEC En.13, n>1],

b) "Use the developer's tools to create selected parts of the TOE and compare with the submitted version of the TOE" [ITSEC En.l 7, n>3].

127 This chapter defines an approach to sampling, with the aim of allowing the ITSEFs to take a consistent approach to planning and costing evaluations. It does not provide a general mandate for sampling, neither does it state how sampling might be carried out. Rather, the aim is to set general boundaries within which acceptable approaches to sampling should lie.

128 Sampling needs to be justified taking into account the possible impact on security of the TOE. The impact depends on what might be missed as a result of sampling. Consideration also needs to be given to the evaluation level, the nature of the deliverables to be sampled, and the requirement not to ignore any security enforcing or relevant functionality.

11.2 Interpretation

129 In effectiveness analysis, sampling is not allowed. Where sampling is allowed, the general principle of sampling is that only the lowest level of representation may be sampled.

11.2.1 Procedural Aspects

130 The approval of the certification body must be obtained if sampling is intended to be used in any area others than those listed in this chapter.

131 An objective of this section is to provide the basis of a sampling plan and sampling procedure for those areas where sampling is explicitly allowed according to this chapter. The following principles must be followed whenever sampling is performed:

- a rationale must be provided, the sample used must be recorded and agreed by the certification body prior to the work commencing (enabling the certification body to ensure sampling consistency across scheme

evaluations); the sample size will depend on a number of TOE dependent factors, but must be as a minimum 20%;

- the sample must be representative of all aspects relevant to the areas sampled, in particular, a selection of tests covering a variety of components, security enforcing and security relevant functions, developer sites (if more than one is involved) and hardware platform types (if more than one is involved); and

- the sponsor and developer must not be informed in advance of the sample.

11.2.2 Sampling the Architecture

132 Sampling from security enforcing or security relevant parts of the architecture is not allowed.

11.2.3 Sampling the Detailed Design

133 Sampling is not allowed within security enforcing parts of the detailed design.

134 At E2, sampling may be possible within security relevant parts of the detailed design. However the evaluators must always examine the detailed design of every security relevant component. Where sampling is applied, it must be done within components, rather than between components. This means that each component still has to be checked for existence, but may be sampled for correctness.

11.2.4 Sampling the Source Code or Hardware Drawings

135 At El and E2, source code or hardware drawings are not a deliverable.

136 At E3, source code or hardware drawings are a deliverable and sampling is possible in the following way. [ITSEC E3.12] requires a “description of correspondence between source code... and the detailed design.” Evaluators must not sample the checking of the existence of the correspondence i.e. they must verify that the description of correspondence is complete and addresses every basic component in the detailed design. However, they may sample the checking of the correctness of the correspondence. Checking the correctness of correspondence may require analysis of some of the source code or hardware drawings.

137 At E4 and above, sampling the correctness of correspondence is not allowed.

138 At E5 and above, the evaluators are required to check that the source code or hardware drawings are “structured into small, comprehensible sections”. This should allow for easier traceability and understanding. The main need is to check that the same technique has been used to sensibly structure the code or drawings, a fact, that can be established without the need to consider every section. This action may therefore be sampled.

139 At E6, the evaluators are required to check that the correspondence “explains the correspondence between the security mechanisms as represented in the source

code... and the formal specification of SEFs in the security target". For this action, sampling is not allowed.

11.2.5 Sampling the Testing

140 As regards tests performed by the developer, the evaluator may sample the checking of these tests. For test coverage evidence, the evaluator may not sample the existence of the evidence but may sample the correctness of the evidence.

141 Should an error be detected in a sample of the developer's tests, the error must be corrected and tests re-run by the developer to confirm the correction of the error and if necessary, additional tests must be run by the developer to demonstrate the absence of side effects [ITSEC En.13, n>2]. Should further errors be found during subsequent samples from the same set of test results, then the certification body must consider the situation on a case-by-case basis as to the impact on the assurance and security of the TOE. In cases where the level of assurance or confidence in the security is brought into question, the TOE must be returned to the developer/ sponsor for a re-run of the tests.

12 Traceability Model

12.1 Background

142 [ITSEC] does not provide a definitive model of the relationship between components, functions and mechanisms. This raises the problem of providing a definitive model for traceability.

12.2 Interpretation

143 This interpretation proposes the [ITSEC] traceability model, however, it is conceivable that other models could also be valid.

144 There is no explicit requirement in [ITSEC] for a stepwise refinement from the security enforcing functions through all levels of representation. However, if it is not possible to demonstrate the correct traceability through all levels of representation, the design documentation would not be a correct representation of the TOE. Also, the traceability through all levels of refinement should correspond to the development and evaluation process.

145 At the requirements phase, the security target specifies security enforcing functions [ITSEC En.2, En.3 and En.4]. Sponsors could also define mandatory security mechanisms [ITSEC 2.4].

146 At the architectural design phase, the target of evaluation is broken down into components of hardware, firmware or software. A clear and effective separation between security-enforcing and other components must be provided at this phase [ITSEC 4.20]. “The specification will distinguish between what the TOE will do (the top level description) and how it will do it (the top level design)” [ITSEC 4.20]. At the architectural design phase, the security enforcing functions defined in the security target are mapped into the security enforcing components [ITSEC En.6]. No specific refinement of the security enforcing functions is required at this level.

147 At the detailed design phase, the architectural design of the TOE is refined. Components of the architectural design are broken down into lower level components. Depending on the development method and complexity of the TOE intermediate levels may exist. [ITSEC 4.21] stipulates that “This phase of development process covers the refinement of the architectural design of the TOE to a level of detail that can be used as a basis for programming and/or hardware construction, i.e. all stages of design and specification below the initial top level specification. Components identified at the lowest level of specification are called basic components”. In the course of refinement, functionality necessary to support security enforcing functions is introduced. These functions are called security relevant functions and the corresponding components are called security relevant components. Starting with E3, the detailed design must provide specifications down to the granularity of basic components. The specifications of basic components

must be detailed enough to be suitable for producing source code or hardware drawings.

148 The following points apply to the diagrams within this topic:

The detailed design may consist of only one level of representation.

At E1, [ITSEC] sets requirements only for the requirements and architectural design phases (for strength of mechanisms, see section 6.5).

At E2, [ITSEC] sets requirements only for the requirements, architectural design and detailed design phases. In particular, there is no requirement for the basic components specification but there is for specifications of mechanisms.

149 The legend of the diagrams within this topic is:

BC = Basic Component
SEF = Security Enforcing Function
SEC = Security Enforcing Component
SRF = Security Relevant Function
SRC = Security Relevant Component
M = Security Mechanism

12.2.1 Logical and Physical Approach

150 [ITSEC] treats logical and physical design as different aspects of the development process. The result of the requirements phase is a logical design in which security enforcing functions are derived from security objectives. The architectural design breaks down the TOE into its major components (physical design). The detailed design includes a logical description of the TOE, including the specification of mechanisms (logical approach), and a physical description of the TOE, i. e. the specification of components through all levels of refinement (physical approach). A link between the logical and physical representation in the detailed design is provided because the specification of a mechanism comprises the mapping to components as well as a specification of interdependencies between these components. Starting with E3, the security enforcing and security relevant components of the TOE are decomposed down to the granularity of basic components. Then, it is possible to trace the representations of the security enforcing functions down to basic components and source code modules / hardware drawings. Implementation is the physical representation of the basic components in source code modules / hardware drawings.

151 The level of specification of security mechanisms is generally at the basic components level.

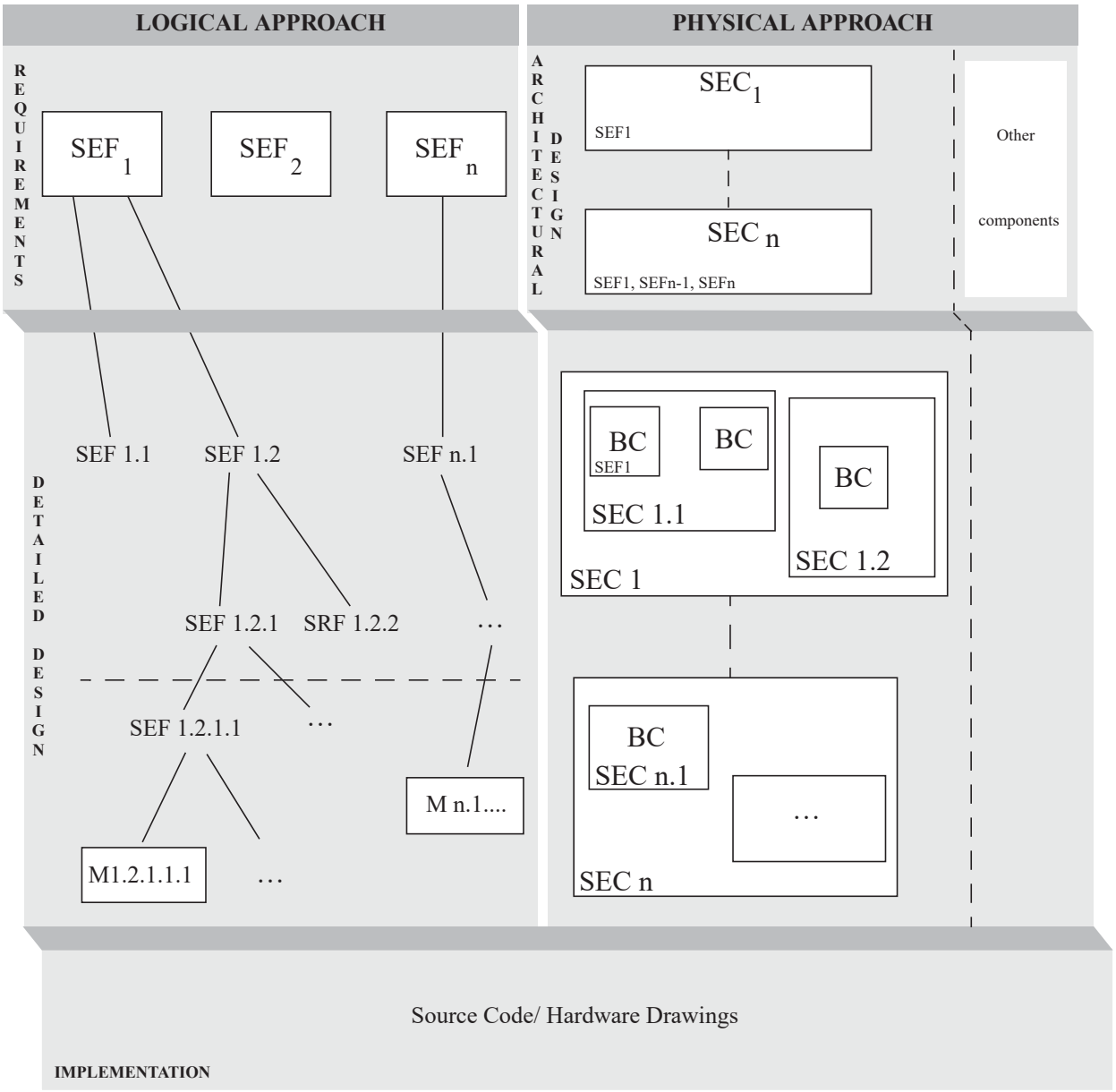

Fig. 12.1 -Logical and physical approach

12.2.2 Approach to Relating the Results of Development Phases

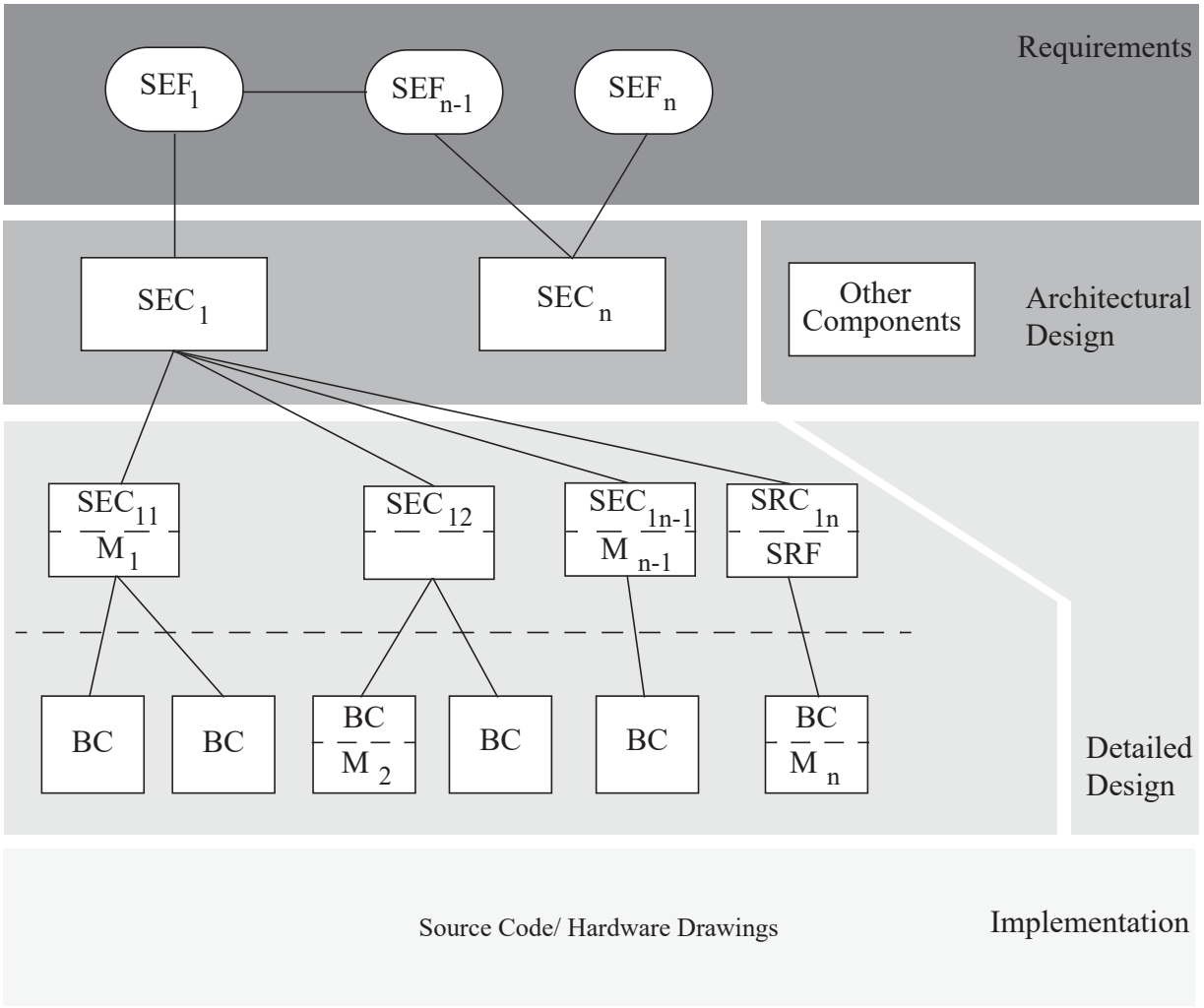

Fig. 12.2 -Approach to relating the results of development phases

13 Semiformal Methods

13.1 Background

152 This topic provides an interpretation on the use of semi-formal methods in TOE documentation.

153 [ITSEC] require the following use of semi-formal notations:

a) the security target shall include a semi-formal description of the SEFs at E4 and E5,

b) a semi-formal notation shall be used in the architectural design to produce a semi-formal description at E4 and E5,

c) a semi-formal notation shall be used in the detailed design to produce a semi-formal detailed design at E4-E6.

154 [ITSEC] give the following characteristics of semi-formal notations:

a) [ITSEC 2.66] states that semi-formal specifications reduce the possibility of ambiguity and imprecision, to which informal specifications are particularly prone;

b) [ITSEC 2.72] states that a semi-formal style of specification requires:

- the use of some restricted notation (or notations);
- the use of the notation shall be in accordance with an informally specified set of conventions included or referenced from the specification; and
- the notation shall allow the specification of both the effect of a function and all exception or error conditions associated with that function.

155 [ITSEC 2.72-2.75] describes and identifies examples of semi-formal styles of specification.

156 [ITSEM] requires the evaluators to check the following for the architectural design:

a) that the notations used, and the manner of their use, are appropriate [ITSEM 4.5.53]; and

b) that the language used is capable of expressing features relevant to security [ITSEM 4.5.55].

157 Note that [ITSEC] does not mandate the use of a semi-formal methodology, though such a methodology would aid in meeting the [ITSEC] traceability requirements.

13.2 Interpretation

158 There are two reasons why [ITSEC] requires use of semi-formal notations at high assurance levels, namely:

a) the evaluators gain a clearer understanding of the TOE design and behaviour; and

b) the likelihood of refinement errors being present in the TOE is reduced.

159 In general, a semi-formal notation should be used in conjunction with explanatory informal text. The semi-formal notation (rather than the informal one) shall be refined correctly between levels. Where a TOE has more than one level of detailed design there shall exist, for each level, a semi-formal detailed design. Traceability through the hierarchy of the semi-formal detailed design shall be possible. For each level, of the semi-formal detailed design, the informal description shall explain the semi-formal detailed design. Any informal description shall be consistent with the semi-formal one.

160 While it is preferable for developers to integrate their semi-formal and informal work, they are allowed to initially do the design work informally and, later produce a semi-formal description. However, any errors found while producing the semi-formal notation shall be corrected, in all notations and in all levels. A developer who chooses to produce a semi-formal description in this way therefore runs the risk of needing to perform significant rework.

161 The semi-formal description shall enumerate all security relevant facts pertinent to the higher level.

14 Covert Channel Analysis

14.1 Background

162 In [ITSEC] the requirements for covert channels are not clearly defined. If due to the nature of the TOE, or some specific functionality (e.g. a one-way function), it is obvious there is a potential vulnerability due to a covert channel, then as a minimum there should be some evidence that the issue has been considered. This will typically be an informal argument as to why the covert channel is not exploitable or why it does not represent a significant risk.

14.2 Interpretation

163 If a security target specifies requirements for covert channels (e.g. it specifies a maximum bandwidth), then work is required under correctness. Should the security target specify specific threats from covert channels, then work is required under effectiveness. However, even if there are no specific statements relating to covert channels, the subject should be still addressed under effectiveness.

164 The following is required for covert channel:

a) a description of known covert channels (both storage and timing); the method used for identifying the covert channels will be described, clearly indicating that the problem has been carefully considered;

b) an estimate of the bandwidth of each identified channel, together with the basis of the estimate; and

c) arguments on the exploitability of each channel in practice, and any remedial measures taken or recommended.

165 At E5, a systematic search for covert channels will need to be demonstrated, together with a thorough approach to the estimation of channel bandwidth and channel exploitability.

166 At E6, a much more rigorous approach is required, for example the use of Kemmerer's shared resource matrix methodology, [ITSEM 4.5.32] to search for covert channels. A more rigorous engineering approach to estimating channel bandwidth will also be required, which in some cases will involve taking actual measurements. Thorough and complete arguments on exploitability will be required. This may require consideration of combinations of channels.

167 It is of secondary importance under which aspects of effectiveness the analysis is performed. [ITSEM] for example, mentions covert channels under both construction vulnerability and binding analysis.

168 The example functionality classes F-B2 and F-B3 of [ITSEC, Annex A] contain requirements on covert channels. These requirements should be addressed under correctness.

169 Where claims are not made in the security target for the maximum bandwidth of any covert channels then the exploitability arguments should take the estimated bandwidth into account. For some TOEs a very small bandwidth covert channel may represent a serious problem. The [TCSEC] guidance states that storage channels of more than 100 bps should not exist, that those between 100 and 10 bps should be audited, and those below 1 bps need not be considered. This guidance should not be taken as default [ITSEC] criteria and should be reconsidered in light of the current technology.

15 Functionality Classes

15.1 Use of Functionality Classes in Security Targets

15.1.1 Background

170 The [ITSEC 2.59-2.64] criteria permit reference within the security target to predefined classes of security enforcing functions (SEFs)[1] called functionality classes.

171 Predefined functionality classes may be referenced for a variety of technical and commercial reasons. They are often claimed by developers when users are expected to be conversant with the [TCSEC] or when they might facilitate comparison between products. They are also specified by users who are required to meet particular computer security standards.

172 Predefined functionality classes are sets of SEFs. Functionality classes permit standardisation and provide the sponsor with a shorthand way of expressing claims for the TOE.

173 A number of example functionality classes have been defined to correspond closely to the functionality requirements of [TCSEC] classes C1 to A1. They are included, as F-C1 to F-B3, amongst the example functionality classes given in [ITSEC Annex A]. It should be noted that the examples there have not been formally validated and their presentation is not suitable for traceability analysis work. None of the functionality classes currently available meet all the requirements of [ITSEC 2.59-2.61].

15.1.2 Interpretation

References to Functionality Classes

174 Where the sponsor wishes to claim a referenced functionality class, the claim should be stated under the specification of the security enforcing functions (The statement may then be conceptually replaced with statements from the referenced class.). The statement should in the form “The TOE shall implement all the security enforcing functions of functionality class F-nn as specified in Reference x”. The claim should clearly state that the TOE meets one or more predefined functionality classes. There should be no need to provide further elaboration, e.g. by incorporation of the text from the [ITSEC] (but see below).

175 The SEFs described by the relevant functionality class will therefore form part or all of the specification of the SEFs in the security target. The TOE may then be evaluated against the security target as expanded by the functionality class.

1. also known as functionality requirements

176 A functionality class shall be referenced as a whole. It is unacceptable to claim that a TOE meets part of a functionality class, or to claim that it meets a functionality class when used in combination with some other product outside the TOE.

177 Wherever possible, functionality classes should be referenced within the ST. Where the sponsor sees a benefit in including the functionality class text, the text shall be copied verbatim to the security target and its origin clearly stated. The reason for the inclusion shall be stated, for instance, it may be to allow traceability by uniquely labelling the individual statements in the functionality class.

178 The sponsor may wish to claim functionality additional to that in the functionality class. The additional functionality shall be expressed separately from the functionality class claim under the normal SEF generic headings of identification and authentication, access control, etc. The sponsor may claim any combination of functionality class and evaluation level.

Effect of Evaluation Level

179 At E1 and E2, the SEFs shall be stated. The informal style of the [ITSEC] example functionality classes is therefore sufficient.

180 At E3 and E4, SEFs shall be described. This requires the sponsor to provide more information than appears in the functionality class; in effect, the sponsor shall describe how the SEFs are to be provided for the particular TOE. This description shall be traceable to the statements in the functionality class.

181 At E5 and E6, the SEFs shall be explained. This interpretation shall be traceable to the functionality class statements.

182 At E4-E5 a semi-formal specification of the SEFs is required in addition to the informal one; at E6, a formal specification is required. Again, these parallel specifications shall be traceable to the functionality class.

ITSEC Example Functionality Classes - Evaluator Actions

183 As the [ITSEC] functionality class examples have not yet been validated, validation work will be required before certification of such claims. Such validation can be undertaken during the evaluator action to check that there are no inconsistencies in the security target. Validation must include:

a) checking the consistency of statements within the class; and

b) checking that TOE-specific class dependencies have been identified, including generic terms such as authorised user, subject and object, which must be defined elsewhere in the security target.

184 Individual statements within the functionality class shall be labelled to facilitate traceability.

185 Any generic problems arising from the use of example functionality classes should be documented and forwarded to the certification body. Any task involving an example functionality class should contact the certification body for a list of known problems and TOE-specific dependencies.

186 The ITSEF is required to report in the ETR on the extent to which the TOE meets the claimed functionality class.

187 Where the security target merely references the functionality class, no additional evaluator effort is necessary. The functionality class is treated as part of the security target and evaluated as normal.

188 Where the security target provides more than a reference to a functionality class, some additional work will be necessary. As a minimum, the evaluators shall check that the sponsor's derived functionality class text in the security target is a correct copy of the [ITSEC] functionality class.

189 Where the sponsor does not follow the guidance in this section, it will be necessary to provide additional evidence that the TOE's SEFs are consistent with the claimed functionality class. In many cases, further guidance will be required from the certification body, and it may not be possible for the certificate/certification report to confirm explicitly that the claimed functionality class has been met. Normally the minimum evidence required will be a cross-reference between statements in the security target and statements in the [ITSEC] functionality class, including interpretations/explanations of any difference in key terms. This evidence shall be presented in a separate annex to the security target.

190 In all cases, the evaluation work programme shall specify how the necessary checks will be performed, and the ETR shall present the results of these checks.

15.2 F-C1 and F-C2 Requirements in Regard to Discretionary Access Control

15.2.1 Background

191 A hierarchical dependency will develop among the operating system specific functionality classes F-C1 to F-B3 as further functional requirements are introduced for each class. For requirements relating to discretionary access control (DAC) this means additional requirements, concerning the granularity of the access rights, that are to be administered by the TOE.

192 In the following paragraphs the hierarchical property of the functionality classes is used to detail the requirements of the various functionality classes. The specification for each class is highlighted by contrasting it with that of a higher functionality class. Additional requirements introduced in a higher functionality class are not applicable to the lower class.

15.2.2 Interpretation

193 **F-C1-Requirements Contrasted with Those of F-C2:**

a) Starting with F-C2 [ITSEC A.12, first sentence], it is required that the TOE identifies and authenticates a user uniquely.

b) Starting with F-C2 [ITSEC A.12, 5th sentence], it is required that the TOE is able to establish the identity of a user for every interaction.

c) Starting with F-C2 [ITSEC A.13, 4th sentence], it is required to be able to grant access rights for an object down to the granularity of an individual user.

d) Starting with F-C2 [ITSEC A.15], accountability for events down to the granularity of an individual user (User Id) is required.

194 Paragraph 193 a) implies that for F-C1 [ITSEC A.8], a unique identification and authentication of an individual user is not yet required. Rather, successful identification and authentication of a user demonstrates being a member of a certain user group.

195 Paragraphs 193 a) to b) imply that a structure of rights based on individual users is not yet required for the functionality class F-C1. Access rights to objects under the administration of rights are granted to user groups. Hence, for the functionality class F-C1, the subjects of the administration of rights are user groups.

196 Paragraphs 193 b) to c) imply that the granularity of user groups as subjects is adequate since accountability and the granting of access rights down to the granularity of an individual user are not required.

197 **F-C2 Requirements Contrasted with Those of F-B2 and F-B3:**

a) The requirements in regard to discretionary access control do not change at F-B1.

b) Starting with F-B2 [ITSEC A.39, 3rd and 4th sentence], it is required to permit granting different access rights to the operator and the system administrator concerning an object of the system administration.

c) Starting with F-B3 [ITSEC A.61, first sentence], this requirement is extended to all subjects and objects under the administration of rights. This results from the requirement of being able to supply for each object which is under the administration of rights, a list of users and user groups with their associated access rights.

198 The F-C2 references in paragraph 193 imply that, for the functionality class F-C2, the subjects of the administration of rights are user groups and individual users.

199 Paragraphs 197 b) and 197 c) imply that it is not required for F-C2 to permit granting each individual user different access rights for an object. It is only required for F-C2 to permit granting an access right to each user.

Discretionary Access Control by Means of the "Protection Bits" Mechanism

200 The realisation of access control by means of the "protection bits" mechanism, as in UNIX operating systems, meets all functional requirements of the functionality class F-C2 with regard to discretionary access control.

201 When realising access control by means of the "protection bits" mechanism, different access rights can be defined for the owner ("owner"), a user group ("group") and all other users ("other"), by setting the protection bits (owner, group, other) of an object. Individual users may belong to multiple user groups. It is therefore possible, to grant access rights down to the granularity of a single user [ITSEC A.13, 4th sentence] by assigning the user to a user group with the relevant access right or by granting the relevant access right to "other". Since it is not required for F-C2 to permit granting different access rights for each individual user to an object under the administration of rights, the "protection bits" mechanism meets all functional requirements of F-C2 with regard to discretionary access control.

16 Generation of the TOE

16.1 Background

202 The production phase is not adequately described in the [ITSEC] and it is not clear what is meant by the terms “manufacturing”, “generation”, “installation”, and “configuration”.

203 The [ITSEC] alludes as follows to:

Installation:

> [ITSEC 2.3]“...managing, purchasing, installing, configuring, operating and using the TOE...”.
>
> [ITSEC 4.30]“...concerned with secure delivery, installation and operational use of a TOE...”.
>
> [ITSEC 4.31]“...to configure the TOE during installation...during installation and configuration at the user’s site.”.
>
> [ITSEC En.28] “... how the system/product shall be installed and how, if appropriate it shall be configured.”.

Configuration (see also Installation):

> [ITSEC 3.33]“...any configuration and installation procedure...”.
>
> [ITSEC 6.16]“**Configuration:** the selection of one of the set of possible combinations of features of a TOE”.

Generation:

> [ITSEC En.32] “...procedures for delivery and system generation shall be stated/described/explained”.
>
> [ITSEC 6.50]“**Production:** the process whereby copies of the TOE are generated for distribution to customers”.

204 **“Installation” and “Configuration”:**

In [ITSEC 4.31, En.28] it is indicated that “installation” and “configuration” may belong together and that sequence between the two activities cannot be fixed. In order to determine a sequence it may be necessary to decide, on a case-by-case basis or the decision may depend on whether the TOE is a product or a system. Further, the question is whether there is always a configuration phase in addition to the installation phase [ITSEC En.28].

205 **“Generation”:**

Considering [ITSEC E.32, 6.50], it is not clear whether generation takes place before or after the delivery.

206 **“Manufacturing”:**

While checking the [ITSEC] for manufacturing/manufacturer it seems that “manufacturing” means the same as “production” or comprises “development” and “production” (see also [ITSEC 6.26] “**Developer**...”).

16.2 Interpretation

207 Considering the sequence of these terms in [ITSEC 2.3] and [ITSEC 4.30] the following figure is proposed:

Phases	Results/actions
Development	Develop master copy of the TOE.
Production	Produce copies of the TOE.
Delivery	Transfer copies of the TOE from the production site to the installation site including intermediate stages.
Installation	Configuration, generation.
Operation	Configuration.

208 **Production**:

Within [ITSEC], development is interpreted as including production [ITSEC 4.23, 4.24]. This means the development environment/process is interpreted as covering the development and production environment/ process.

Examples are:

The title “Construction - The Development Environment” is interpreted as “Construction - The Development and Production Environment”.

The title “Aspect 3 - Developer Security” is interpreted as
“Aspect 3 - ***S***ecurity of Development and Production Environment”.

209 [ITSEC En.21 and En.22, n>1]:

> The term “development environment” is interpreted as “development and production environment”.

210 **Manufacturing:**

211 [ITSEC En.16, n>1]:

> The term “manufacturing process” is interpreted as “development and production process”.

212 **Installation, Generation and Configuration:**

213 [ITSEC Aspect 1]:

> The title “Aspect 1 - Delivery and Configuration” is interpreted as “Aspect 1 - Delivery and Installation”.

214 In [ITSEC En.32, En.34], the term “generation” is used. This term is always interpreted as “installation”.

215 [ITSEC En.32, n>1] is interpreted as:

> If different configurations are possible, the impact of the configurations on security shall be (stated/described/explained). The procedures for delivery and installation shall be (stated/described/explained). A procedure approved by the national certification body for this evaluation level shall be followed, which guarantees the authenticity of the delivered TOE. While installing the TOE, any configuration options and/or changes shall be audited in such a way that it is subsequently possible to reconstruct exactly how the TOE was initially configured and when the TOE was installed.

216 [ITSEC En.34, n>1] is interpreted as:

> The sentence “Search for errors in the system generation procedures” is interpreted as “Search for errors in the installation procedures”.

217 In [ITSEC En.35] the impact of different configurations must be taken into consideration for secure start-up and operation.

17 Hardware TOE

17.1 Background

218 [ITSEC] has been designed to address both software and hardware TOEs.

219 [ITSEC 1.2] stipulates that "these criteria have been designed so as in the main part to be equally applicable to technical security measures implemented in hardware, software or firmware. Where particular aspects of evaluation are intended only to apply to certain methods of implementation, this is indicated as part of the relevant criteria."

220 The [ITSEC] concepts are directly applicable to hardware TOEs, but its application may be different.

221 When applying the criteria to hardware TOEs, some aspects of correctness and effectiveness require interpretation. This is especially true for aspects of the architectural design, detailed design, binding analysis and the strength of mechanism analysis.

17.2 Interpretation

222 When applying the [ITSEC] to hardware TOEs, two types can be considered:

- type 1: TOEs composed of separate identifiable physical units (e.g. PC cards)
- type 2: TOEs which are physically one integrated circuits (IC) but may contain one or more logically separate units (e.g. Smart card ICs).

223 The distinction between these two types of TOE may assist the evaluator in subsequent interpretation and application of [ITSEC] to hardware TOEs.

17.2.1 Requirements

224 The interpretation concerning the scope and boundaries of the TOE as stated in JIL section 4.1 applies. No specific interpretations are required for hardware TOEs.

225 The technical and technological properties of hardware TOEs should be specified as security enforcing functions preferrably at an abstract level, independent of implementation details and to be taken into account during the evaluation.

17.2.2 Architectural Design

226 [ITSEC 4.20] stipulates that the "Architectural design covers the overall top level definition and design of the TOE".

227 This phase of the development process is essential to define the major components of the TOE, the basic structure of the TOE, its external interfaces and for E2 and above its separation between major “hardware and software” components.

228 For type 1 TOEs it is acceptable to map the major architectural components to the physical devices performing specific functions (e.g. CPU, RAM, ROM, Bus and I/O chips). For type 2 TOEs a refinement step is necessary, to map major architectural components to an IC's internal logical devices.

229 The [ITSEC En, n>1] criteria related to the separation between security enforcing components and the others may be achieved in a type 1 hardware TOE by using different hardware devices with a specified interface (e.g. a device composed of an IC, a battery and a communication component).

230 For type 1 and 2 hardware, the mapping of the security enforcing functions to physical components in order to identify the security enforcing components may not be easy to do (e.g. which component really does process the SEF, the CPU or a CPU in conjunction with its associated memory and bus?). Due to this difficulty in mapping, it may be easier for some hardware TOEs to consider all TOE components as security enforcing components.

231 [ITSEC En.6, n>3] requires that “The architectural design shall describe/explain how the chosen structure provides for largely independent security enforcing components”. The objective of this requirement is to minimise the interrelationships between the TOE security enforcing components and the others. Due to the physical nature of the TOE, the interfaces between components are fully specified, particularly the possible interfaces between the security enforcing components and the others. The level of interdependency must be justified in order to satisfy the requirements, in accordance with the chapter 2 interpretation on the level of rigor ("describe/explain").

232 [ITSEC En.6, n>4] requires that “The architectural design shall explain why the interrelationships between the security enforcing components are necessary”. The objective of this requirement is to minimise any interrelationships between the security components in order to facilitate testing. Each interface between security components must be fully justified.

233 As for the definition of the supporting protection mechanisms implemented in hardware or firmware, the following explanations are provided. For hardware TOEs it is important to distinguish between internal mechanisms (identified in the detailed design) from those mechanisms external to the TOE which shall be identified as supporting protection mechanisms.

17.2.3 Detailed Design

234 There are two ways of designing hardware TOEs:

- through a classical process of hardware drawing: the development process depends essentially on the technologies used (specific method and tools)

and can be described by refining architectural design into a sufficient level of detail to implement the TOE;
- through a hardware description language (HDL) : the detailed design being similar to software.

17.2.4 Implementation

235 Typically, the two main steps in testing a hardware TOE are:

- the “TOE prototype” tests (which are called “characterisation tests”),
- the acceptance tests performed on each TOE at the end of the production phase.

236 The characterisation tests can be considered to provide evidence for the correct implementation of security enforcing functions and mechanisms. The evaluators shall check the developers manufacturing process has appropriate acceptance tests to confirm and verify the correct operation of the TOE and the components of which it is constructed during its manufacture. In order to check the results of characterisation and acceptance tests where specialist test equipment is essential the evaluator may have to witness and verify the tests rather than personally perform the tests.

237 Timing should be considered when testing Hardware TOEs. It is acceptable to use simulation to support testing.

238 Hardware drawings or HDL statements corresponds to source code for software TOE.

17.2.5 Configuration Control

239 JIL Chapter 9 interpretations apply, in particular for the production phase.

240 [ITSEC En.17 n>3] requires the evaluator to “use the developers tools to rebuild selected parts of TOE and compare with the submitted version of the TOE”. At the design level, this can be achieved using the relevant tools where tools have been used to build parts of the TOE, otherwise the evaluators can witness and verify TOE construction. At implementation level, this is covered by the development and production visit (see JIL section 8.1).

17.2.6 Programming Languages and Compilers

241 [ITSEC 4.25] explicitly stipulates that this [ITSEC] aspect only applies to software and firmware TOEs. The [ITSEC En.18 En.19 n>2] requirements can be extrapolated to type 2 TOEs where they have been developed using “silicon compilers” (e.g. by HDL method) which uses compiling tools and supporting libraries during the develop of the TOE. In addition, the technology used for the implementation has to be specified. At E4 and above, corresponding to the requirements for compilers, the parameters of the technology used have to be documented.

17.2.7 Developers Security

242 JIL chapters 8 and 16 have already given interpretations applying to these types of TOEs (the development and production sites are considered by this aspect).

17.2.8 User Documentation and Administration Documentation

243 There is no specific hardware interpretation for these aspects.

17.2.9 Delivery and Configuration

244 The interpretations given in the JIL chapter 16 for the TOE Generation, and in the JIL chapter 10 for the TOE delivery apply.

17.2.10 Start-up and Operation

245 [ITSEC En.35 n> 1] requires that "if the TOE contains hardware which contains security enforcing hardware components, then administrator, end-user, or self initiated diagnostic tests shall exist that can be performed on the TOE in its operational environment."

246 The TOE shall contain diagnostic tests for security enforcing hardware components. For certain TOEs this activity may only be possible at specific stages in the TOE life-cycle, especially in type 2 hardware.

17.2.11 Suitability of Functionality

247 There is no specific hardware interpretation for this aspect as assessment of suitability of the TOE remains the same.

17.2.12 Binding of Functionality

248 The analysis of potential links between security enforcing functions and security enforcing mechanisms needs to consider design solutions as:

- physical links between components (e.g. electrical connection),
- and some dynamic or time effect which could cause conflict or dysfunction of the security enforcing functions and mechanisms.

249 At E4 and above the implementation phase has to be taken into consideration when performing the binding analysis: the effectiveness of the functional solutions depends on the implemented technology.

17.2.13 Strength of Mechanisms

250 Strength of mechanism analysis is applicable to hardware TOEs but its analysis may be complicated by the TOE's underlying technology.

251 In hardware TOEs, a critical mechanism is usually implemented by more than one basic component.

252 The following particular types of attack can be identified:

- attacks involving physical modifications of the internal TOE structure: they generally bypass SEFs and the composition of the different components which implement the mechanisms have to be taken into consideration, so such attacks should be considered under the aspect of vulnerability analysis (see section 17.2.15).

- attacks without physical modifications of the internal TOE structure:

  a) these may be similar to traditional direct attacks on mechanisms implemented on software but may involve physical means, or

  b) may be attacks on the mechanisms implementing technical and technological properties of the TOE (see section 17.2.1).

  These are appropriate for strength of mechanism analysis.

17.2.14 Ease of Use

253 There is no specific hardware interpretations for these aspects.

17.2.15 Construction and Operation Vulnerability Assessment

254 Hardware TOEs, both type 1 and type 2, can be subject to vulnerabilities which can be exploited by physical tampering of the TOE. Such tampering could circumvent the effectiveness of security enforcing functions. This aspect must be considered during vulnerability assessment and penetration testing.

255 The evaluator must consider whether any special tools and or techniques can be used to tamper with the TOE to exploit a weakness. If so, such tools and techniques shall be considered. The vulnerability analysis will be concerned with resources and expertise needed to exploit vulnerabilities. The minimum strength of mechanisms scale shall be used accordingly (see JIL section 6.4).

18 Binding Analysis

18.1 Background

256 This topic seeks to promulgate a better understanding of the requirements of binding analysis. This topic provides an interpretation of the [ITSEC] binding analysis requirements and provides a technique which can be used in the production of a binding analysis.

257 Binding analysis is concerned with vulnerabilities in the construction of the TOE which, if exploited by indirect (JIL section 6.3) attack, would prevent the TOE from meeting one or more of its security objectives. If the TOE is to successfully defend itself against indirect attack then its security functions must support each other where necessary; also the security functions must not conflict with each other. In practice this involves consideration of the potential interactions, between security functions, which are provided for and permitted by the construction of the TOE.

18.2 Interpretation

258 [ITSEC 3.17-3.19] indicate that binding analysis is concerned with "security enforcing functions and mechanisms". Therefore binding must also be demonstrated for lower levels of representation than that of the security enforcing functions, in accordance with the requirements of the target assurance level as given by [ITSEC Figure 4].

259 Binding is undertaken on the assumption that all security enforcing functions are correctly realised. Therefore, correct binding of the security enforcing functions will automatically provide a degree of assurance for the correct binding of the lower level representations.

260 It must be recognised however that, for complete two-way traceability of security enforcing functions to exist, not only must the full security functionality exist at all levels of representation, but additional functionality (which could invalidate higher level binding) must not be introduced at lower levels. Note that, whilst this becomes increasingly significant at higher assurance levels, it is in part consistent with stricter design criteria such as "It shall be structured into well-defined, largely independent basic components" [ITSEC E4.8] and "Unnecessary functionality shall be excluded from security enforcing and security relevant components" [ITSEC E5.8]. Realistically however, additional detail of potential significance to binding can be expected to exist at lower representational levels and must be considered.

18.2.1 Bases for Binding at E1 and E2

261 At E1 and E2 [ITSEC Figure 4] does not mandate consideration of the Detailed Design, in which the majority of mechanisms will be identified. In this case it would be appropriate to consider binding of the security enforcing functions specified in

the security target and any supporting mechanisms identified in the Architectural Design. The Architectural Design must also be considered to take account of any detail significant to binding.

262 [ITSEC Figure 4] does not exclude additional material being included in the basis for effectiveness analyses. At E2 therefore, where Detailed Design is required for correctness, it would still be possible to base the binding analysis on mechanisms, if this was considered to give the clearest binding model.

18.2.2 Bases for Binding at E3 and above

263 At assurance levels E3 and above it is considered that the mechanisms identified during correctness offer the clearest basis for a binding analysis.

264 In many cases a mechanism-based binding analysis will need to be supplemented by significant detail from lower level representations (i.e. from low level physical design components identified in detailed design and from implementation components), in accordance with the target assurance level as given by [ITSEC Figure 4]. This concerns specifically the full traceability of binding interactions, and ensuring that binding is not invalidated by low level effects.

265 Where a given mechanism implements a number of security enforcing functions then the binding, within the mechanism, of the security enforcing functions must be considered. At the mechanism level this can be achieved by confirming that the mechanism is integrated and self consistent, with reference to the various implementational requirements of the security enforcing functions. Where lower level representations are consulted it may be possible to identify distinct components which have been traced from the different security enforcing functions and confirm the binding of such components.

18.2.3 Security Relevant Considerations

266 [ITSEC 3.17-3.19] refers to “security enforcing functions and mechanisms”. Sensibly this must be extended to cover security relevant functions and mechanisms in the following categories (in this chapter, security enforcing and relevant functions and mechanisms are referred to as binding elements):

a) mechanisms which enforce binding;

b) binding elements with capability to access secure data which lies outside their immediate implementational scope, e.g. in the absence of appropriate protection mechanisms;

c) binding elements which process secure data that must subsequently be purged;

d) binding elements which are assigned privilege; the interactions facilitated by each privilege mode must be considered.

267 Binding must be completed by justifying the ability of the TOE to withstand any indirect attack, misuse or error originating from a user, an application or an external "device". The emphasis should be on justifying the ability of security enforcing and relevant components as whole to withstand such activity, considering sufficiently representative types of such external agents.

18.2.4 Interactions

268 [ITSEC 3.18] requires binding to analyse "all potential interrelationships" between binding elements. The interaction which exists between a given pair of binding elements might be multiple. A sequence of related interactions can usually be considered sensibly as a single, compound interaction. Where two interaction sequences are unrelated however, it is clearest to consider them as distinct interactions.

269 In addition to considering the effect of direct interactions between binding elements it is also necessary to consider the consequent indirect interactions; i.e. if binding element 1 interacts directly with binding element 2, and if binding element 2 interacts directly with binding element 3, then element 1 interacts indirectly with element 3. In general there is then scope for further indirect interactions.

270 All potential interactions must be considered, including any which are not required for correctness of the implementation. It may be possible to reduce the number of potential interactions to be considered whilst maintaining rigour, by identifying interaction types, each of which is representative, in all significant respects, of a number of interactions.

18.2.5 Source Code Analysis

271 For aspects concerning Source Code Analysis specifically, refer to JIL section 7.2.2.

18.2.6 Covert Channel Analysis

272 For aspects concerning Covert Channel Analysis specifically, refer to JIL topic 14.

19 Formal Methods

19.1 Background

273 [ITSEC 2.81-2.83] provides explanations about formal model of security policy and [ITSEC 2.76-2.78] about formal specification.

274 [ITSEC 2.78] provides examples of formal notations. Additional notations are CSP, VSE, B. More detailed information can be found in [EWICS TC7 Guidance].

275 The use of formality as applied to the [ITSEC] deliverables is described as follows:

- At E4 and above, a formal model of security policy is required with an informal interpretation of this model in terms of the security target [ITSEC En.1, En.2 n>3]; referred to within this topic as FMSP and its informal interpretation;
- at E6, a formal description of the architecture of the TOE shall be provided [ITSEC E6.1 - E6.5]; referred to within this topic as FAD;
- at E6, a formal specification of security enforcing functions is required [ITSEC E6.1, E6.2]; referred to within this topic as Formal SEFs.

276 The above requirements apply to all types of TOEs i.e., software, hardware or firmware products or systems.

277 The following figure gives an overview of [ITSEC] requirements.

	FMSP	FAD	Formal SEF
E4	X		
E5	X		
E6	X	X	X

Fig. 19.1 -Required use of formal notation

19.2 Interpretation

19.2.1 FMSP

278 The FMSP's aim is to enhance the assurance by formally specifying and proving that the TOE correctly enforces the stated security policy.

279 As described in [ITSEM 6.B.25], the system security policy for a system, or the product rationale for a product, should state in the security target the important principles of security (referred to as “the Security Policy”):

- for a system, it corresponds to the security objectives defined in the System Security Policy (SSP) which shall be addressed by a combination of TOE Security Enforcing Functions and personnel, physical or procedural means associated with the system, as described in [ITSEC 2.9];

- for a product, it corresponds to the product rationale which gives an equivalence to the “system security objectives” by identifying the product's security features and all environmental assumptions [ITSEM 6.B.25-6.B.28]. In some cases, a product rationale may specify security objectives.

280 At E4 and above, part or all the TOE Security Policy of the system or product, known in ITSEC as the Underlying Security Policy, shall be expressed in a formal style in the FMSP.

19.2.2 Formal SEFs

281 [ITSEC E6.2] requires to provide a formal specification of the SEFs. There is no specific interpretation for this aspect.

19.2.3 FAD

282 The FAD must be traceable to the semi-formal detailed design and source code/ hardware drawings, that if necessary informal correspondence between them can be easily understood.

283 The FSMP and FAD must be separated by a significant design step. Sufficient design steps, described in a formal language, may include the step from abstract behaviour description to a concrete description or flattening a distributed structure into a global structure with constraints.

284 Examples of insufficient design steps include the implementation of trivial constraints or simple data representation changes, such as implementing a set as a sequence.

19.2.4 Relationship between FMSP, formal SEFs, FAD

285 The following figure summarises the relationship between FMSP, formal SEFs and FAD.

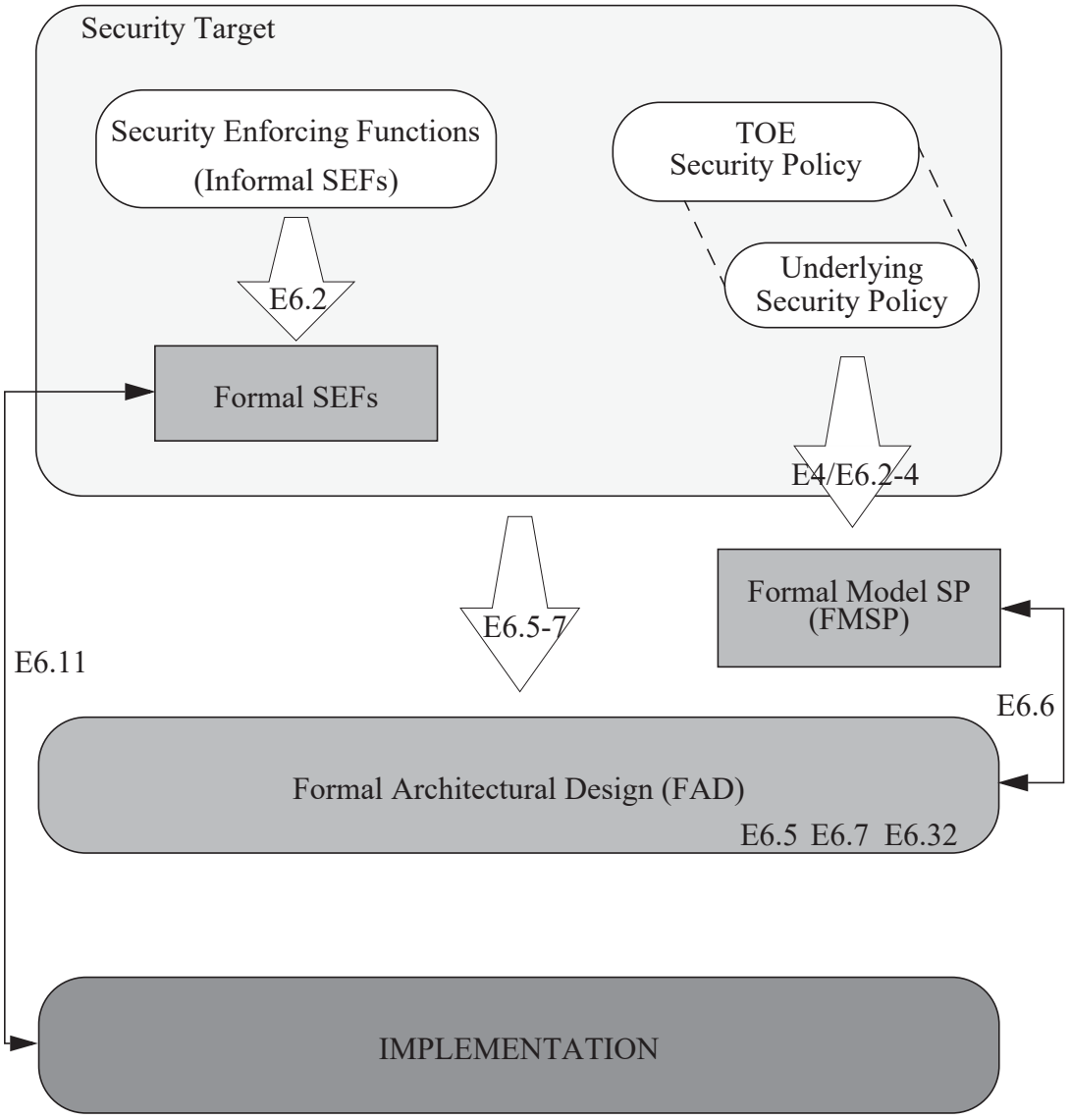

Fig. 19.2 -Relationship between FMSP, Formal SEFs and FAD

19.2.5 Proofs

286 The [ITSEC] requires evidence in order to satisfy requirements. The following proofs shall be presented as evidence.

287 **FMSP proofs**

FMSP proofs shall prove evidence for the correctness of the security model. This includes but is not limited to the internal consistency of the security model, in the sense of non-existence of contradictions and invariance (i.e. the impossibility of transition from secure to insecure states) of its properties.

288 **SEF proofs**

SEF proofs shall prove evidence for the correctness of the SEF. This includes but is not limited to the internal consistency of the SEF, in the sense of non-existence of contradictions and invariance of its properties.

289 **FAD proofs**

FAD proofs shall prove evidence for the correctness of the architectural design. This includes but is not limited to the internal consistency of the architectural design, in the sense of non-existence of contradictions and invariance of its properties.

290 A proof must provide evidence that establishes the validity of the subject being proved. It shall be accompanied by a justification of why the proof obligation is a correct formal statement for the subject being proved.

291 Proofs must be formal and independently checkable. It must be possible for someone other than the author to check the correctness of the proof. This may be done in one of four ways:

- Manual proof, checked by a different human reviewer,
- Manual proof, checked by an automated proof checker,
- Computer generated proof, checked by a human reviewer,
- Computer generated proof, checked by an automated proof checker.

292 Proofs to be checked by a human reviewer must be well structured, give intuitive explanations for proof steps, and make good use of lemmas. It is often inappropriate to perform all steps of a proof; however, any steps left out for the reviewer must be obvious and clearly derivable, in that it must not require creative proof work to generate them. Experience has shown that such a level of formality is achievable.

20 Ease of use

20.1 Context of Ease of Use

20.1.1 Background

293 [ITSEC 6.31] defines ease of use as “an aspect of the assessment of the effectiveness of a TOE, namely that it cannot be configured or used in a manner which is insecure but which an administrator or end-user would reasonably believe to be secure.”

294 [ITSEM 5.8.73] states “This aspect of effectiveness investigates whether the TOE can be configured or use in a manner which is insecure but which an administrator or end-user of the TOE would reasonably believe to be secure.”

295 The aim of the ease of use analysis is thus to demonstrate that the TOE cannot operate in an insecure state without the user being aware of this; otherwise an operational vulnerability will exist.

20.1.2 Interpretation

296 When performing an ease of use analysis the consequences of hardware failure must be considered for completeness.

Identification of operational states

297 Operational states should be identified through consideration of TOE interfaces and procedures.

298 All modes of operation of the TOE shall be considered.

299 Assessment of whether a particular operational state is insecure is made with reference to the security target.

Identification and Analysis of Insecure States

300 Where it is apparent that an error could lead to a potentially insecure state, consideration should be given as to whether:

a) the TOE or documentation gives a clear warning;

b) the TOE prevents entry to the insecure state; or

c) another facility will detect and report the insecurity.

Annex A References

- ITSEC: Information Technology Security Evaluation Criteria, Version 1.2 June 1991,

- ITSEM: Information Technology Security Evaluation Manual, Version 1.0 September 1993,

- TCSEC: Trusted Computer Systems Evaluation Criteria, DOD 5200.28-STD Department of Defence, United States of America, December 1985.

- EWICS TC7: Guidance on the use of Formal Methods in the Development and Assurance of High Integrity Industrial Computer Systems, S O Anderson, R E Bloomfield, G L Cleland (eds.).