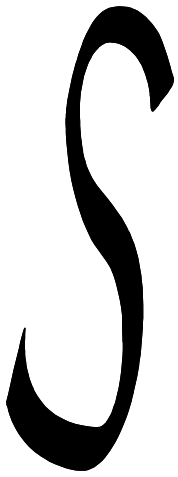
COMMISSION OF THE EUROPEAN COMMUNITIES
DIRECTORATE GENERAL XIII: Telecommunications, Information Market and Exploitation of Research
DIRECTORATE B: Advanced Communications Technologies and Services
B6: Security of Telecommunications and Informations Systems



Information Technology Security Evaluation Manual

(ITSEM)

Version 1.0

Address: Rue de la Loi 200, B-1049 Brussels - Office Av. de Beaulieu 9 2/11 B-1160 Brussels
Telephone: direct line (+32 2) 296 36 02, Secretariat 296 34 19 - Facsimile (+32 2) 296 62 71

Electronic Mail: dhe@postman.dg13.cec.be - ISCO @postman.dg13.cec.be Compuserve 1000137.1072

Contents

This page is intentionally left blank

Part 0 Introduction

Contents

Chapter 0.1 Introduction

Background

0.1.1 In May 1990 France, Germany, the Netherlands and the United Kingdom published the *Information Technology Security Evaluation Criteria* [ITSEC] based on existing national work in their respective countries. After widespread international review the ITSEC has been developed in two further versions of which the current version 1.2 is the basis for this document.

0.1.2 An important reason for wishing to produce these international harmonised criteria was that such harmonisation is one of the prerequisites of international mutual recognition of the certificates which summarise the results of Information Technology (IT) security evaluations and confirm that the evaluations have been properly carried out. It is also a prerequisite of mutual recognition that the methods used to apply these harmonised criteria should themselves be harmonised. On completion of the ITSEC therefore, the four countries continued to co-operate, with the aim of agreeing a common approach to the conduct of IT security evaluations, at least to the extent necessary to provide the required confidence to facilitate mutual recognition.

0.1.3 Much work had already been done and some of this published on the development of IT security evaluation methods. In the UK this included CESG Memorandum Number 2 [CESG2], developed for government use, and the "Green Books" series of the Department of Trade and Industry, including V23-Evaluation and Certification Manual [DTI23], for commercial IT security products. In Germany, the German Information Security Agency published their IT Evaluation Manual [GISA1].

0.1.4 The basic approach was to harmonise existing security evaluation methods in each of the four countries to the extent necessary to ensure that national evaluation methods conform to a single philosophy. It was initially felt that the work should be limited to harmonisation of existing methods. However, it has been necessary to extend the existing work and to develop some new ideas in order to achieve these objectives.

General Scope

0.1.5 This IT Security Evaluation Manual (ITSEM) builds on the ITSEC Version 1.2, describing how a Target Of Evaluation (TOE) should be evaluated according to these criteria. The specific objective of the ITSEM is to ensure that there exists a harmonised set of evaluation methods which complements the ITSEC.

0.1.6 The ITSEM is a technical document, aimed predominantly at partners in evaluation (primarily evaluators but also sponsors and certifiers), but it is also of interest to vendors, developers, system accreditors and users. It contains sufficient detail of evaluation methods and procedures to enable technical equivalence of evaluations performed in different environments to be demonstrated. The document will be freely available. The ITSEM will apply to evaluations carried out both in commercial and government sectors.

0.1.7 For the purposes of mutual recognition it is necessary that some parts of the ITSEM be prescriptive on evaluators. However most of the ITSEM is descriptive or intended to provide guidance.

0.1.8 In order to put the evaluation methods prescribed and described into a context, it is necessary to include in the ITSEM some outline information on certification and how it may be organised.

0.1.9 This document stresses the importance of independence of evaluation from any commercial pressures from a sponsor or developer of a TOE. However first party evaluation, in the sense of evaluation performed by another part of the sponsoring or developing organisation, is not precluded provided that the requirements of the national scheme are fulfilled.

0.1.10 The ITSEM has been written from the perspective that certification follows the evaluation. The case that an evaluation is followed by a supplier's declaration is outside the scope of this document although, even in this case, use of the ITSEM may still be helpful.

Structure and Content

0.1.11 The rest of this document consists of six parts, one of which has annexes. Each part has been written from the perspective of the targeted audience for that part. Some subjects are handled in more than one part, but with a different level of detail.

0.1.12 Part 1 of the ITSEM describes an IT security framework providing background and rationale for IT security, evaluation, certification and system accreditation. This part is of a general nature. It is intended for a management audience.

0.1.13 Part 2 of the ITSEM gives basic information on the establishment and running of an evaluation and certification scheme, describing the general features of the certification process and the organisation of it. It is of interest to those wishing to understand the certification process.

0.1.14 Part 3 of the ITSEM explains the evaluation philosophy which underlies the ITSEC. It contains the principles which must be followed by the evaluators when evaluations are performed. It gives further explanation and clarification of the ITSEC concepts to provide a better basis for understanding the technical issues underlying evaluation.

0.1.15 Part 4 of the ITSEM is the key part for those closely involved in evaluation. All mandatory text is in this part. It gives an overview of how evaluation is performed and describes evaluation in terms of input, process, output. However it does not provide guidance for all details of evaluation.

0.1.16 Part 5 of the ITSEM provides examples of the application of ITSEC, demonstrating how the ITSEC can be applied to the evaluation of systems and products.

0.1.17 Part 6 of the ITSEM gives guidance on evaluation to sponsors, vendors, developers, system accreditors and users. It is particularly concerned with preparing the inputs, and using the outputs, from evaluation.

Numbering and Text Conventions

0.1.18 Each paragraph within a part is uniquely identified by the combination of part number, chapter number and paragraph number within the chapter. The first use of an ITSEM glossary term within a part is shown in bold type. Italics are used to signify emphasis or quotation. In part 4 of the ITSEM prescriptive text has been highlighted by shading and bolding complete sentences or paragraphs.

Further Developments

0.1.19 The ITSEC version 1.2 is currently being used for a trial period. During this period it is expected that improvements to the ITSEC will be proposed in the light of practical experience. The ITSEM also draws upon other documents ([CESG2], [DTI23], [GISA1]) that have already been extensively discussed and used in practice in national schemes; it is considered that the ideas and concepts have been carefully balanced and that the structure chosen for the document is the right one for maximum consistency and usability.

0.1.20 The current version of the ITSEM benefits from significant revisions arising from widespread international review. The review process has been assisted by the Commission of the European Communities who organised an international workshop in September 1992, at which version 0.2 was discussed. This event was supplemented by written comments and contributions from reviewers, which the authors have sought to take into account in preparing version 1.0. It is recognised by the authors of the ITSEM that in some are as of the ITSEM detailed guidance is still lacking, but that where appropriate, additional information in those areas will appear in later versions, as both this document and the ITSEC evolve in line with experience.

Chapter 0.2 Background Information

Points of Contact

0.2.1 Comments and suggestions are invited, and may be sent to any of the following addresses, bearing the marking "ITSEM Comments":

Commission of the European Communities
DIRECTORATE GENERAL XIII: Telecommunications, Information Market and Exploitation of Research
DIRECTORATE B: Advanced Communications Technologies and Services
Rue de la Loi 200
B-1049 BRUSSELS
Belgium

For France:

Service Central de la Sécurité des Systèmes d'Information
18 Rue du Docteur Zamenhof
F-92131 ISSY LES MOULINEAUX

For Germany:

Bundesamt für Sicherheit in der Informationstechnik
Postfach 20 03 63
D-53133 BONN

For the Netherlands:

Netherlands National Comsec Agency
Bezuidenhoutseweg 67
P.O. Box 20061
NL-2500 EB THE HAGUE

National Security Service
P.O. Box 20010
NL-2500 EA THE HAGUE

For the United Kingdom:

Head of the Certification Body
UK IT Security Evaluation and Certification Scheme
Certification Body
PO Box 152
CHELTENHAM
Glos GL52 5UF

Glossary and References

0.2.2 The glossary contains definitions of technical terms that are used with a meaning specific to this document. Technical terms used within this document that are not defined here are used throughout this document in a manner consistent with the ITSEC glossary. If they are neither defined here or in the ITSEC then they are used with their generally accepted meaning.

Abbreviations

0.2.3	ANSI	-	American National Standards Institute
0.2.4	CAD	-	Computer Aided Design
0.2.5	CASE	-	Computer Aided Software Engineering
0.2.6	CB	-	Certification Body
0.2.7	CRC	-	Cyclic Redundancy Check
0.2.8	DAC	-	Discretionary Access Control
0.2.9	ETR	-	Evaluation Technical Report
0.2.10	EWP	-	Evaluation Work Programme
0.2.11	FMEA	-	Failure Mode and Effects Analysis
0.2.12	FMSP	-	Formal Model of Security Policy
0.2.13	I&A	-	Identification and Authentication
0.2.14	IPSE	-	Integrated Project Support Environment
0.2.15	ISO	-	International Standards Organisation
0.2.16	ITSEC	-	Information Technology Security Evaluation Criteria
0.2.17	ITSEF	-	Information Technology Security Evaluation Facility
0.2.18	ITSEM	-	Information Technology Security Evaluation Manual
0.2.19	MAC	-	Mandatory Access Control
0.2.20	MARION	-	Méthode d'Analyse de Risques Informatiques et d'Optimisation par Niveau
0.2.21	MELISA	-	Méthode d'Evaluation de la Vulnérabilité Résiduelle des Systèmes
0.2.22	MMI	-	Man Machine Interface
0.2.23	PCB	-	Printed Circuit Board

0.2.24	PDL	-	Program Description Language
0.2.25	PID	-	Personal Identification Device
0.2.26	PIPSE	-	Populated Integrated Project Support Environment
0.2.27	SSADM	-	Structured Systems Analysis and Design Methodology
0.2.28	SEISP	-	System Electronic Information Security Policy
0.2.29	SEF	-	Security Enforcing Function
0.2.30	SoM	-	Strength of Mechanisms
0.2.31	SPM	-	Security Policy Model
0.2.32	SSP	-	System Security Policy
0.2.33	TCB	-	Trusted Computing Base
0.2.34	TOE	-	Target of Evaluation
0.2.35	T&T	-	Technique and Tool

Glossary

0.2.36 **Asset**: information or resources to be protected by the technical and non-technical countermeasures of a TOE

0.2.37 **Audit Trail**: the set of records generated by a TOE in response to accountable operations, providing the basis for audit

0.2.38 **Authentication**: the verification of a claimed identity

0.2.39 **Binding Analysis**: the determination of whether the totality of security enforcing functions, together with the description of their inter-working as described in the architectural design, fulfils the totality of security objectives, i.e. covers all threats enumerated in the security target

0.2.40 **Certificate/Certification Report**: the public document issued by a CB as a formal statement confirming the results of the evaluation and that the evaluation criteria, methods and procedures were correctly applied; including appropriate details about the evaluation based on the ETR

0.2.41 **Certification Body**: a national organisation, often the National Security Authority, responsible for administering ITSEC evaluations within that country

0.2.42 **Construction Vulnerability**: vulnerabilities which take advantage of some property of the TOE which was introduced during its construction

0.2.43 **Correct Refinement**: the refinement of a function described at one abstraction level is said to be correct if the totality of effects described at the lower abstraction level at least exhibits all the effects described at the higher abstraction level

0.2.44 **Countermeasure**: a technical or non-technical security measure which contributes to meeting the security objective(s) of a TOE

0.2.45 **Deliverable**: an item or resource that is required to be made available to the evaluators for the purpose of evaluation

0.2.46 **Error**: a failure to meet the correctness criteria

0.2.47 **Evaluation Technical Report**: a report produced by an ITSEF and submitted to the CB detailing the findings of an evaluation and forming the basis of the certification of a TOE

0.2.48 **Evaluation Work Programme**: a description of how the work required for evaluations is organised; that is it is a description of the work packages involved in the evaluation and the relationships between them

0.2.49 **Exploitable Vulnerability**: a vulnerability which can be exploited in practice to defeat a security objective of a TOE

0.2.50 **Impact Analysis**: an activity performed by a sponsor to determine if a re-evaluation of a changed TOE is necessary

0.2.51 **Impartiality**: freedom from bias towards achieving any particular result

0.2.52 **Information Technology Security Evaluation Facility**: an organisation accredited in accordance with some agreed rules (e.g. [EN45]) and licensed by the CB to perform ITSEC security evaluations

0.2.53 **Information Technology Security Evaluation Manual**: a technical document containing sufficient detail of evaluation methods and procedures to enable mutual recognition

0.2.54 **National Scheme**: a set of national rules and regulations for evaluation and certification in accordance with the ITSEC and ITSEM

0.2.55 **Object**: a passive entity that contains or receives information

0.2.56 **Objectivity**: a property of a test whereby the result is obtained with the minimum of subjective judgement or opinion

0.2.57 **Operational Vulnerability**: vulnerabilities which take advantage of weaknesses in non-technical countermeasures to violate the security of the TOE

0.2.58 **Potential Vulnerability**: a suspected vulnerability which may be used to defeat a security objective of a TOE, but the exploitability or existence of which has not yet been demonstrated

0.2.59 **Problem Report**: a concise report, produced by the evaluators, sent to the CB outlining an error, a potential or actual vulnerability in the TOE

0.2.60 **Re-evaluation**: an evaluation of a previously evaluated TOE after changes have been made

0.2.61 **Re-use**: the use of previous evaluation results when one or more previously evaluated components are incorporated into a system or product

0.2.62 **Repeatability**: a repeated evaluation of the same TOE to the same security target by the same ITSEF yields the same overall verdict as the first evaluation (e.g. E0 or E5)

0.2.63 **Representation**: the specification of a TOE at a particular phase of the development process (one of requirements, architectural design, a level of detailed design, implementation)

0.2.64 **Reproducibility**: evaluation of the same TOE to the same security target by a different ITSEF yields the same overall verdict as the first ITSEF (e.g. E0 or E5)

0.2.65 **Subject**: an active entity, generally in the form of a person, process, or device [TCSEC]

0.2.66 **Suitability Analysis**: the determination that the security enforcing functions described in the security target are able to act as countermeasures to the threat(s) identified in the security target (suitability is only assessed at this level)

0.2.67 **Vulnerability**: a security weakness in a TOE (for example, due to failures in analysis, design, implementation or operation).

References

0.2.68 The following books and papers are referenced in the text:

BDSS Risk Quantification Problems and Bayesian Decision Support System Solutions, Will Ozier, Information Age, Vol. 11, No. 4, October 1989.

BOE Characteristics of Software Quality - TRW North Holland, B.W. Boehm, Software Engineering Economics - Prentice Hall, 1975.

CESG2 Handbook of Security Evaluation, CESG Memorandum No. 2, Communications-Electronics Security Group, United Kingdom, November 1989.

CRAMM CCTA Risk Analysis and Management Methodology, Guidance on CRAMM for Management, Version 2.0, CCTA, February 1991.

DTI23 Evaluation and Certification Manual, V23 Department of Trade and Industry, United Kingdom, Version 3.0, February 1989

ECMA A Reference Model for Frameworks of Computer-Assisted Software Engineering Environments, ECMA TR/55.

EN45 General Criteria for the Operating of Testing Laboratories, EN 45001.

GASSER Building a Secure Computer System, Morrie Gasser, Van Nostrand Reinhold.

GISA1 IT Evaluation Manual, GISA 1990.

GISA2 IT Sicherheitshandbuch, BSI 7105, Version 1.0, March 1992.

GUI25 General Requirements for the Technical Competence of Testing Laboratories, International Standards Organisation, ISO Guide 25, 1982.

ISO65A Software for Computers in the Application of Industrial Safety Related Systems, ISO/IEC JTC1/SC27 N381, November 1991.

ITSEC Information Technology Security Evaluation Criteria - Harmonised Criteria of France, Germany, the Netherlands, and the United Kingdom, Version 1.2, June 1991.

LINDE Operating System Penetration, R Linde, Proceedings of the AFIPS, NCC, pp 361-368, 1975.

MCC Factors in Software Quality, J A McCall, General Electric n.77C1502, June 1977.

MS1629A Procedures for performing a failure mode, effects and criticality analysis, MIL-STD-1629A, US DoD, November 1980.

NIS35 Interpretation of Accreditation Requirements for IT Test Laboratories for Software and Communications Testing Services, NAMAS Information Sheet NIS35, NAMAS Executive, National Physics Laboratory, United Kingdom, November 1990.

OSI OSI Basic Reference Model, Part 2 - Security Architecture, ISO 7498 (1988(E)).

PCTE Portable Common Tool Environment Abstract Specification (December 1990; ECMA 149).

PCTE+ Portable Common Tool Environment (Extended) Definition Team Final Report (14 December 1992).

SRMM Shared Resource Matrix Methodology: An Approach to Identifying Storage and Timing Channels, R A Kemmerer, ACM Transactions on Computer Systems, Vol. 1, No. 3, August 1983.

TCSEC Trusted Computer Systems Evaluation Criteria, DoD 5200.28-STD, Department of Defense, United States of America, December 1985.

TNI Trusted Network Interpretation of the TCSEC, National Computer Security Center, United States of America, NCSC-TG-005, Version 1, 31 July 1987.

TDI Trusted Database Interpretation of the TCSEC, National Computer Security Center, United States of America, NCSC-TG-021, April 1991.

This page is intentionally left blank

Part 1 IT Security Framework

Contents

Figures

Chapter 1.1 Introduction

Assets, Threats, Risks, Confidence and Countermeasures

1.1.1 Information Technology (IT) has become essential to the effective conduct of business and the affairs of state, and is becoming increasingly important to the affairs of private individuals affected by the use of IT. Information is something to be gained and protected in order to advance one's business or private affairs, and should therefore be regarded as an **asset**. The importance of such assets is usually expressed in terms of the consequential damage resulting from the manifestation of threats. Damage may be caused directly or indirectly, by disclosure, improper modification, destruction or abuse of information. Risk increases with the size of the likely damage and the likelihood of the threats being manifested.

1.1.2 The information in IT systems has to be protected against threats which lead to harmful impacts on assets. Threats can be deliberate (e.g. attacks) or inadvertent (e.g. mistakes or failures).

1.1.3 In order to reduce risk, specific **countermeasures** will be selected. These countermeasures will be physical, personnel, procedural or technical in nature. *Technical countermeasures* or *IT countermeasures* are the security enforcing functions and mechanisms of the IT system; *non-technical countermeasures* or *non-IT countermeasures* are the physical, personnel and procedural countermeasures. ITSEC evaluation is principally concerned with technical countermeasures.

1.1.4 The primary security objective of an IT system is to reduce the associated risks to a level acceptable to the organisation concerned. This can be achieved by security functions and features of the IT system.

1.1.5 The confidence that may be held in the security provided by the IT system is referred to as assurance. The greater the assurance, the greater the confidence that the system will protect its assets against the threat with an acceptable level of residual risk.

1.1.6 The higher the ITSEC evaluation level and strength of mechanisms, the greater the assurance the user can have in the countermeasures built into the IT system or product. The evaluation level required by a user depends on the acceptable level of known residual risk and can only be determined by means of a threat and risk analysis for a specific case. Security and costs have to be balanced. Products or systems with higher evaluation levels will usually be more expensive, as the costs for development and evaluation are likely to increase with increasing evaluation level. Guidance on how to determine an evaluation level as a function of environmental parameters is given in, for example, [GISA2]. Specific advice can be sought from the national organisations mentioned in part 2 of the ITSEM.

Processes in the IT Security Framework

1.1.7 Several processes contribute to the goal of IT security. These are illustrated in figure 1.1.1.

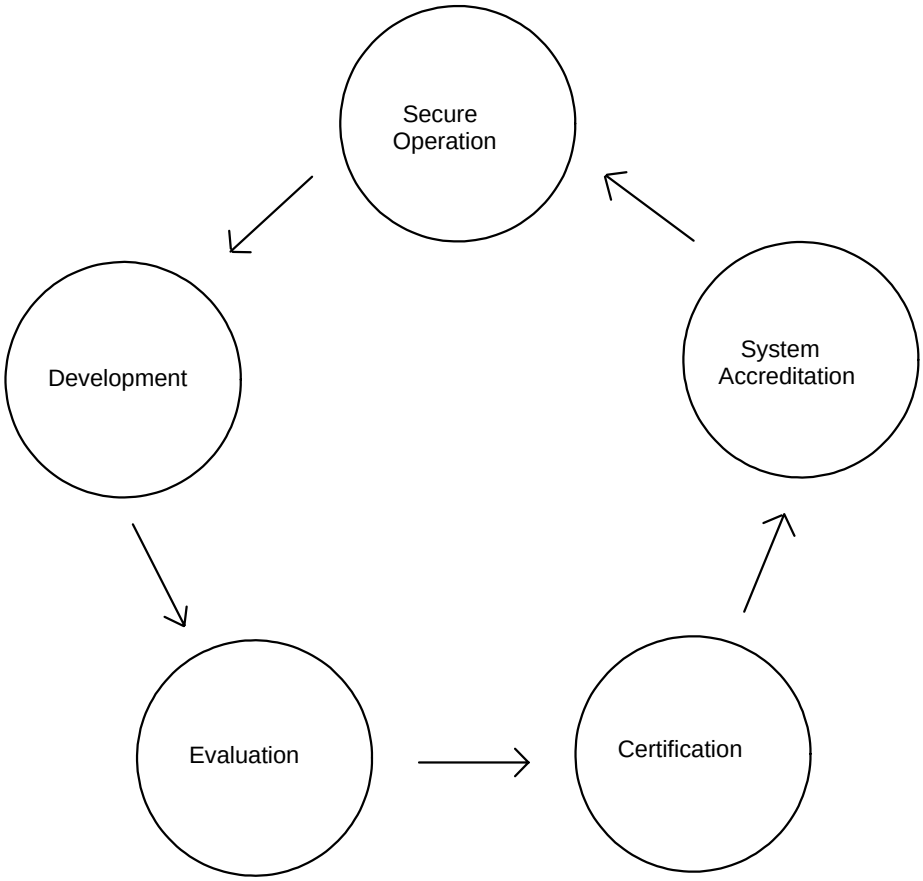


Figure 1.1.1 Processes in the IT Security Framework

1.1.8 This figure shows the idealised context into which IT security evaluation and certification are embedded. The arrows in the figure indicate that one process provides input for another. The processes may be partially interleaved. The sequence of processes is likely to be iterative and permanently ongoing.

1.1.9 In the development process an IT system or product is built. In the evaluation process it is assessed against defined security evaluation criteria. In the certification process it is confirmed that the results of an evaluation are valid and the evaluation criteria have been applied correctly. In the system accreditation process it will be confirmed that the use of an IT system is acceptable within a particular environment and for a particular purpose. In the secure operation process an accredited system is operated according to approved procedures, but changes to the environment may require changes which provide input to the development process.

1.1.10 The term accreditation is used in two different contexts. Accreditation of an IT security evaluation facility (ITSEF) is the procedure for recognising both its **impartiality** and its technical competence to carry out evaluations. Accreditation of an IT system (as defined in the introduction of the ITSEC) is a procedure for accepting an IT system for use within a particular environment. System accreditation is concerned with IT and non-IT countermeasures, whereas the ITSEM is principally concerned with IT countermeasures. System accreditation is outside the scope of the ITSEC/ITSEM.

Context of Evaluations

1.1.11 The context of evaluations has three aspects:

a) criteria;

b) methodology;

c) **national schemes**.

1.1.12 The criteria represent the scale against which the security of an IT system or product may be measured for its evaluation, development, and procurement. The criteria state what has to be evaluated. The methodology advises how the evaluation should be performed on the basis of the criteria. The national schemes provide organisational rules for the processes of evaluation, certification, and laboratory accreditation in terms of roles, procedures, rights, and obligations. The criteria are contained in the ITSEC and the associated methodology in the ITSEM, only to a level of detail to facilitate mutual recognition between national schemes. Issues of the national schemes are addressed in part 2 of the ITSEM and in the scheme documentation of individual countries.

Chapter 1.2 Evaluation and Certification Process

Basic Concepts

1.2.1 The evaluation process outlined in this chapter is a framework which describes organisational and procedural aspects of the conduct of an evaluation. There are many matters surrounding an evaluation which are treated differently in different nations for reasons of e.g. jurisdiction or national security. The rules of the national scheme take precedence in each of the countries. Matters of national schemes are addressed in part 2 of the ITSEM.

1.2.2 When performed as commercial activities, evaluations according to the ITSEC are subject to economic conditions of the IT market. They must be commercially feasible, i.e. affordable and timely. This goal has to be balanced against the benefits of the evaluation. Principles guiding the evaluation and certification process are presented in part 3 of the ITSEM.

1.2.3 In this type of evaluation:

a) sponsors can set the objectives for the evaluation process;

b) resources from the ITSEF can be made available at the sponsor's request, and;

c) maintenance of **certificates/certification reports** through **re-evaluation** is easily supported.

Involved Parties

1.2.4 The following parties are directly involved in the evaluation process:

a) the sponsor of an evaluation;

b) the developers of an IT product or system;

c) the IT Security Evaluation Facility (ITSEF);

d) the **Certification Body** (CB).

1.2.5 Other parties concerned with evaluation and certification are users and system accreditors. They are mainly concerned with procurement and secure operation.

1.2.6 Figure 1.2.1 indicates that all parties involved look at the evaluation and certification process from a different angle depending on their roles.

1.2.7 The ITSEM contains descriptive information and guidance for sponsors, developers, ITSEFs, system accreditors and certification bodies, and additionally in part 4 of the ITSEM prescriptive information is provided to the evaluators.

1.2.8 The sponsor of an evaluation is the party which initiates and funds the evaluation. In the case of a system evaluation it is likely that the sponsor and system accreditor will be the same organisation.

1.2.9 The ITSEF performs the evaluation, usually as a commercial activity. The evaluation, in accordance with the ITSEM and ITSEC, includes detailed examination of a TOE, searching for **vulnerabilities** and determining the extent to which the TOE's security target is met by its implementation.

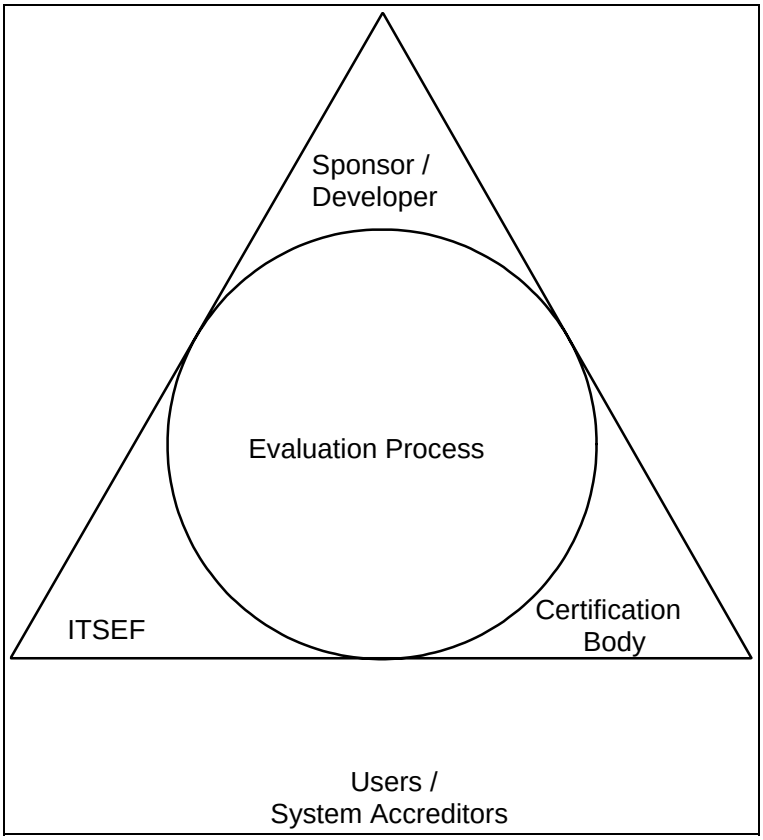


Figure 1.2.1 Parties involved in, or concerned with, evaluation and certification

1.2.10 The independence of the ITSEF from commercial or other pressures from the sponsor or developer of a TOE is regarded as a key issue. However this does not preclude self evaluation or first party evaluation, in the sense of the evaluation being performed by another part of the sponsoring or developing organisation, provided that the requirements of the national scheme are fulfilled.

1.2.11 It is likely that evaluation work is performed under a non-disclosure agreement with the sponsor or developer. An ITSEF should preserve commercial confidentiality.

1.2.12 Another important issue is the impartiality of the ITSEF with regard to evaluations performed. National schemes may impose requirements to be fulfilled by an ITSEF.

1.2.13 The CB checks whether the results of an evaluation are valid and whether the evaluation criteria have been applied correctly. This is to ensure uniformity and correctness of evaluation procedures according to the ITSEM and the ITSEC, and the consistency and compatibility of evaluation results. The CB prepares and issues the certificate/certification report for those TOEs which are found to meet their security target and therefore fulfil the requirements of Chapter 5 of the ITSEC.

1.2.14 The certificate/certification report will be published. Information on their format and content is provided in part 2 of the ITSEM.

Phases of the Evaluation Process

1.2.15 The evaluation process is divided into three phases:

a) Phase I Preparation;

b) Phase II Conduct;

c) Phase III Conclusion.

1.2.16 The process is outlined here for a typical evaluation. In practice there are a number of options, in particular when the evaluation is performed in parallel with the development process. The three phases are described in more detail in part 4 of the ITSEM.

1.2.17 Phase I includes the initial contact between the sponsor and an ITSEF, any feasibility study and preparation for the evaluation. The feasibility study is optional but is particularly recommended for sponsors and developers without prior evaluation experience. The feasibility study will confirm that the sponsor and the developer are well prepared for the conduct of an evaluation and will at least involve a review of the security target. When a successful evaluation seems to be feasible, a list of required evaluation **deliverables**, a plan for their delivery, and an **evaluation work programme** are established. It is sensible to contact the CB to establish a schedule agreed by the sponsor, the developer, the ITSEF, and the CB.

1.2.18 A contract between the sponsor and an ITSEF is normally signed during Phase I. The contract takes account of national regulations and usually includes a non-disclosure agreement.

1.2.19 The evaluation work programme for a specific TOE is based on the deliverables, their delivery schedule and the ITSEC criteria. The required work is divided into evaluation activities to be performed by evaluators according to a time schedule. The task of developing the evaluation work programme is similar to the task of planning in the software development life cycle. No fixed order of applying the criteria is prescribed in the ITSEC, but some sequences of activities are more reasonable and efficient than others. Details on this issue are provided in part 4 of the ITSEM.

1.2.20 Phase II is the main part of the evaluation process. The evaluators perform the ITSEC evaluator actions. This includes penetration testing based on the list of **potential vulnerabilities** and other testing. The **Evaluation Technical Report** (ETR) is prepared in this phase.

1.2.21 In Phase III the ITSEF will provide the final output of the evaluation process, the ETR, to the sponsor/developer, and to the CB as basic input to the certification process. Obligations for confidentiality may require different handling. As the ETR contains sensitive information, the ETR is not a public document and is subject to the rules of the national scheme. It might become necessary for the ITSEF to provide technical support to the CB concerning the ETR.

1.2.22 Certification is outlined in part 2 of the ITSEM.

Problem Handling

1.2.23 Problems identified by the ITSEF during an evaluation are usually discussed between the sponsor, developer and the ITSEF. In the case of severe problems advice should be sought from the CB. If problems cannot be solved, the sponsor may decide to abandon the evaluation. The rules of the national scheme apply in all cases.

Concurrent and Consecutive Evaluations

1.2.24 An evaluation might be performed after development of the TOE has been completed, which is called *consecutive evaluation*, or in parallel with the development of the TOE, which is called *concurrent evaluation*.

1.2.25 The main difference between concurrent and consecutive evaluations is the availability of the various **representations** of the TOE provided as deliverables. In a consecutive evaluation all deliverables required by the ITSEC, from the security target to the operational TOE, are normally available right from the start of the evaluation. In a concurrent evaluation the deliverables will be provided by the sponsor/developer as the development progresses. Concurrent evaluations provide the opportunity for sponsor/developers to react rapidly to problems discovered.

1.2.26 The difference between the two types of evaluations does not have any technical impact, but affects the organisation of an evaluation, i.e. the evaluation work programme. In the concurrent evaluation the order and the time scale of the evaluation activities has to be oriented towards the delivery of the deliverables. Penetration and other testing cannot be performed before the operational TOE is available. The potential consequences of delays and iterations need to be considered.

Product and System Evaluations

1.2.27 According to the ITSEC a product is *a package of IT software and/or hardware, providing functionality designed for use or incorporation within a multiplicity of systems*, and a system is *a specific IT installation, with a particular purpose and operational environment*.

1.2.28 Product and system evaluations are similar; both might be concurrent or consecutive. The main difference concerns the security target. In the case of a system the environment is known and the threats or security objectives are real and can be specified in detail. In the case of a product the threats or security objectives have to be assumed by anticipating the operational purpose and environment of the product, and can only be expressed in generic terms.

Re-evaluation and Re-use of Evaluation Results

1.2.29 When a product or system is evaluated and certified, the certificate/certification report applies to the evaluated version and configuration only. It is likely that security requirements and evaluated products or systems will be subject to change. The certificate/certification report may not apply to the changed product or system and re-evaluation may be required. This is covered in more detail in part 6, annex 6.D.

1.2.30 During the process of re-evaluation it may be desirable to **re-use** the results of the previous evaluation of the TOE. This issue is addressed in part 4, chapter 4.6 and part 6, annex 6.F.

Part 2 Certification Schemes

Contents

Chapter 2.1 Introduction

2.1.1 The ITSEC Version 1.2, in addressing the Certification Process at paragraph 1.31, states: *In order for these criteria to be of practical value, they will need to be supported by practical schemes for the provision and control of independent evaluations, run by appropriately qualified and recognised national* ***certification bodies****. These bodies will award certificates to confirm the rating of the security of TOEs, as determined by properly conducted independent evaluations.*

2.1.2 Such schemes, by ensuring that the methodology set out in the ITSEM is applied consistently and correctly in assessing TOEs against the ITSEC, are a pre-requisite for international mutual recognition of the certificates awarded by their certification bodies.

2.1.3 Provided that all schemes ensure the compliance of evaluations with the ITSEM in performing the evaluator actions set out in the ITSEC, it should be possible to accept that the result from an evaluation under one scheme will be the same as that obtained under any other scheme.

Chapter 2.2 Standards

2.2.1 International and European Standards (ISO Guide 25 [GUI25] and EN45001 [EN45]) have been established to provide general guidance for accreditation and operation of test laboratories. These standards create a framework for the objective testing of all types of products, not simply those in the IT field. Where IT security evaluation and certification are concerned, compliance with these standards is a worthwhile goal, particularly because it would make it easier for any mutual recognition agreement to be recognised by the European Committee for IT Testing and Certification (ECITC).

2.2.2 However, there are factors peculiar to IT security which may make compliance to every detail of the letter of such standards undesirable or difficult to achieve. Therefore, interpretations of EN45001 for the area of IT security evaluation are being prepared in different countries to become part of the national regulations for evaluation, certification and evaluation facility accreditation and licensing, respecting the spirit of those standards. Even with the interpretations, there are some aspects of IT security evaluation which a certification body has to oversee in order to ensure comparable evaluation results.

Chapter 2.3 Formation of ITSEFs

2.3.1 It is essential that evaluations are conducted by evaluation facilities experienced in IT security and in the ITSEC/ITSEM methodology. These facilities should therefore seek compliance with the requirements of EN45001 and their appropriate interpretations with respect to IT security. However, a formal accreditation procedure to this standard is not mandated. This has led to the definition of ITSEF licensing schemes in some countries which have added requirements, especially IT security aspects, to accreditation based on EN45001. These additional requirements are outside of the scope of ITSEM and are not considered further. However these aspects are either not relevant to mutual recognition or would be expected to be the subject of any mutual recognition agreements.

2.3.2 Some national accreditation or licensing schemes are already installed and subject to the above mentioned requirements. Facilities interested in further information, especially on such additional requirements, should seek advice from the appropriate national body (see chapter 2.7).

Chapter 2.4 Evaluation and Certification: Objectives and Benefits

2.4.1 The main objective of certification is to provide independent confirmation that evaluations have been properly carried out in accordance with the approved criteria, methods and procedures and that the conclusions of the evaluations are consistent with the facts presented. Within the context of a scheme operated by a single certification body, this in turn helps to provide grounds for confidence that different evaluation facilities belonging to the scheme are operating to the same standards and that the conclusions of any two evaluation facilities will be equally reliable. Important aspects for these grounds for confidence are summed up in four principles:

a) **Impartiality:** All evaluations must be free from bias.

b) **Objectivity:** The property of a test whereby the result is obtained with the minimum of subjective judgement or opinion.

c) **Repeatability:** The repeated evaluation of the same TOE to the same security target by the same ITSEF yields the same overall verdict as the first evaluation.

d) **Reproducibility:** The evaluation of the same TOE to the same security target by a different ITSEF yields the same overall verdict as by the first ITSEF.

2.4.2 There are a number of benefits in the evaluation and certification process by which the various partners in the process stand to gain. Some of these are outlined below:

a) The Vendor/Developer/Sponsor gains from evaluation and certification in that:

- customers are aware that a successful independent third party evaluation has agreed with the claims made in respect of the product;
- an appropriate certificate will permit entry and acceptability in specialised markets;
- certified products may be used as building blocks for certified systems;
- certification also provides a statement about the quality of a product or system and its development;

b) Users/System Accreditors gain from evaluation and certification in that:

- they can have confidence that a third party assessment has confirmed the security claims of a vendor;
- a certificate provides a useful basis for comparison between different products;
- they get guidance to ensure that the secure configuration of a certified TOE is not compromised;

c) The evaluators gain from evaluation and certification in that:

- they benefit from a wider customer base;
- the independent oversight by the CB provides guidance to the evaluators to ensure they fulfil their obligations;

d) The Certification Schemes benefit in that they allow for:

- comparison, development and maintenance of international standards;
- measurement of internal standards against a set of international criteria;
- encouragement of sponsors by providing wider markets for their products.

Chapter 2.5 The Certification Scheme

2.5.1 The main aims of a certification body are, firstly, to create the conditions under which the work of all the ITSEFs in a scheme will be accurate and consistent and their conclusions valid, repeatable and reproducible and, secondly, in the case of individual evaluations, to provide independent confirmation that they have been carried out in accordance with approved criteria, methods and procedures. In order to fulfil these aims, the CB must perform the following functions (among others):

a) authorise the participation of ITSEFs in the scheme, ensuring compliance with the requirements of the scheme in question;

b) monitor the performance of ITSEFs and their adherence to, and application of ITSEC and ITSEM, issuing additional guidance as necessary;

c) monitor every evaluation carried out by an ITSEF;

d) review all evaluation reports to assess the implications of the results for security and to ensure that they conform to the ITSEC and ITSEM;

e) produce certification reports;

f) publish certificates and certification reports.

2.5.2 All these activities are carried out within the context of a certification scheme. The all-important consistency of standards (and so of the validity and reliability of results) between different ITSEFs can be achieved only in the context of such a scheme. Such consistency is important not only from the point of view of the individual customer and his confidence in an evaluation (and so in the product or system evaluated), but also because it is a prerequisite for achieving international mutual recognition.

2.5.3 Among other functions covered by a certification scheme are: the definition of types of products and systems which can be evaluated; the release of certificates and certification reports and their subsequent maintenance (including the prevention of abuse); the issuing of documents relating to the scheme and its operation; and other aspects of the scheme's day-to-day administration.

2.5.4 Any special national requirements imposed in particular schemes are outside the scope of the ITSEM. Those who require further information on particular **national schemes** should seek advice from the appropriate body among the points of contact listed in chapter 2.7.

Chapter 2.6 Contents of Product Certificates/Certification Reports

2.6.1 Certificates and certification reports will be publicly available.

2.6.2 Certificates and certification reports should at least contain:

a) **Introduction**:

- preliminary material as handled by the national scheme;

b) **Summary**:

- the identity of the ITSEF;
- the identifier of the Target of Evaluation (TOE) including issue number/release number;
- the evaluation identifier assigned by the certification body;
- a summary of the main conclusions of the evaluation;
- the identity of the developer (including sub-contractors as applicable);
- the identity of the sponsor;
- actual evaluation level achieved;

c) **Product Overview**:

- a description of the evaluated configurations;
- hardware description;
- firmware description;
- software description;
- documentation description;

d) **The Evaluation**:

- a brief description of the security target, including a description of the Target of Evaluation security features;
- reference to the **Evaluation Technical Report**;
- the identity of the IT security evaluation facility;
- summary of the main conclusions of the evaluation;

e) **Certification**

- scope of Certificate (e.g. any limits on the application of the Target of Evaluation).

Chapter 2.7 List of Contacts

2.7.1 Following is a list of contacts who can provide advice on evaluation and certification:

For France:

Service Central de la Sécurité des Systèmes d'Information
18 Rue du Docteur Zamenhof
F-92131 ISSY LES MOULINEAUX

For Germany:

Accreditation Body:

Bundesamt für Sicherheit in der Informationstechnik
Referat II 4
Postfach 20 03 63
D-53133 BONN

Certification Body:

Bundesamt für Sicherheit in der Informationstechnik
Referat II 3
Postfach 20 03 63
D-53133 BONN

For the Netherlands:

National Security Service
P.O. Box 20010
NL-2500 EA THE HAGUE

Netherlands National Comsec Agency
Bezuidenhoutseweg 67
P.O. Box 20061
NL-2500 EB THE HAGUE

For the United Kingdom:

Head of the Certification Body
UK IT Security Evaluation and Certification Scheme
Certification Body
PO Box 152
CHELTENHAM
Glos GL52 5UF

This page is intentionally left blank.

Part 3 Philosophy, Concepts, and Principles

Contents

Figures

Chapter 3.1 Introduction

3.1.1 This part describes the evaluation philosophy which underlies the ITSEC, and introduces basic principles of the evaluation work. It presents concepts and notions used in the evaluation process. It provides the technical basis for the national evaluation and certification schemes (part 2 of the ITSEM) and the evaluation process (part 4 of the ITSEM). The principles will be discussed in detail and implemented in part 4 of the ITSEM.

Chapter 3.2 General Philosophy of Evaluation

Confidence and Assurance

3.2.1 The main goal of evaluation is to gain confidence that the TOE satisfies its security target. The evaluation provides a particular degree of confidence that there are no **exploitable vulnerabilities**. The benefits provided by the security objectives in the security target are not assessed during evaluation as these depend on the particular application of the TOE.

3.2.2 The degree of confidence gained by an evaluation depends on the evaluation level and strength of mechanisms. The higher the evaluation level, the larger is the amount of relevant information provided and used, the greater is the effort required for evaluation, and the higher is the resulting assurance. Thus, evaluation may be regarded as a single but complex measurement performed with a degree of accuracy characterised by the evaluation level. Consequently, the more one has to rely on **countermeasures** provided by a TOE, e.g. to reduce a high risk to an acceptable level, the higher should be the evaluation level and the strength of mechanisms. There is an increased likelihood that the TOE will behave as expected and counter the threats adequately.

3.2.3 Assurance in the security provided by a product or system is derived both from examining the product or system and its **representations**, and from understanding the process by which it was developed.

3.2.4 The greatest contribution to assurance derives from examination of representations of the product or system itself. A developer accredited to a quality standard such as ISO 9001 is more likely to be able to produce adequate representations, but such accreditation can in no way substitute for any part of the evaluation.

Repeatability, Reproducibility, Impartiality, and Objectivity

3.2.5 In the context of IT security evaluation and certification, as in the fields of science and testing, ***repeatability***, ***reproducibility***, ***impartiality***, and ***objectivity*** are considered to be important principles.

3.2.6 An evaluation is repeatable, if the repeated evaluation of the same TOE to the same security target by the same ITSEF yields the same overall verdict as the first evaluation.

3.2.7 An evaluation result is reproducible, if the evaluation of the same TOE to the same security target by a different ITSEF yields the same overall verdict as by the first ITSEF.

3.2.8 An evaluation is performed impartially, if the evaluation is not biased towards any particular result.

3.2.9 An evaluation is performed objectively if the result is based on actual facts uncoloured by the evaluators' feelings or opinions.

3.2.10 These four principles are enforced by a certification body within a national scheme. In particular, the certification body ensures that the repeatability and reproducibility of test results is extended to the evaluation result as a whole.

Understanding

3.2.11 The evaluation criteria describe the evidence an evaluation sponsor/developer must produce, and the points the evaluators must check. The evaluation is based on the information provided by the sponsor/developer. Assurance gained by evaluation depends on knowledge about the TOE and its behaviour. The more appropriate and complete the information is about the TOE, the better one is able to understand the TOE. The result is greater confidence that the TOE satisfies its security target. These facts are reflected by the ITSEC requirements for the sponsors/developers to provide construction phase **deliverables** as a set of specifications of the TOE at different levels of abstraction.

3.2.12 Evaluation consists of a combination of observation, theory and experiment. Understanding the TOE is the first prerequisite of good evaluation work. Understanding is gained by assessing the security target and the other deliverables with regard to the correctness criteria. On the basis of their understanding of the TOE and its security target, the evaluators can investigate the TOE with regard to the effectiveness criteria, i.e. whether the TOE can behave in any way contrary to the requirements of the security target or whether the TOE is vulnerable to anticipated threats.

3.2.13 In general, TOEs are far too complex for testing alone to prove that they meet their security target. Exhaustive testing is not feasible. Therefore, evaluation confidence is achieved by evaluators understanding the TOE through analysis of documentation about its construction and operation as well as through testing. There will always remain some doubt about the conformance of the TOE with its security target. There can never be complete assurance but only the evidence of an increased likelihood that the TOE satisfies its security requirements. In general it is desirable to minimise the residual uncertainty. The higher the evaluation level is, the better the evaluators need to understand the TOE.

Modularisation and Software Engineering Principles

3.2.14 Modularisation and other software engineering principles like information hiding etc. usually provide a good starting point for supporting and limiting the necessary evaluation work. Such principles contribute to identifying **potential vulnerabilities**. A well-documented development using well-defined notations will ease the understanding of the TOE by the evaluator. Programming languages with well-defined syntax and semantics are an example relevant to the implementation phase. Development based on good software engineering practice will ease the evaluators' work.

Evaluation Process

3.2.15 A well-defined evaluation method clearly understood by all parties should be employed. Individual evaluation processes for particular TOEs are developed based on the ITSEC criteria, the evaluation philosophy and principles, the **national scheme**, and the evaluation processes described in part 4 of the ITSEM (see figure 3.2.1). The evaluation process should be standardised in order to make monitoring and comparison of results easy and efficient. In practice, the evaluation method is implemented in the **evaluation work programme** and the performance of the identified evaluator activities. The enormous variety of possible security targets and TOEs precludes a detailed prescription. Elaboration of the evaluation method is described in part 4 of the ITSEM, and in national schemes.

Fehler! Textmarke nicht definiert.

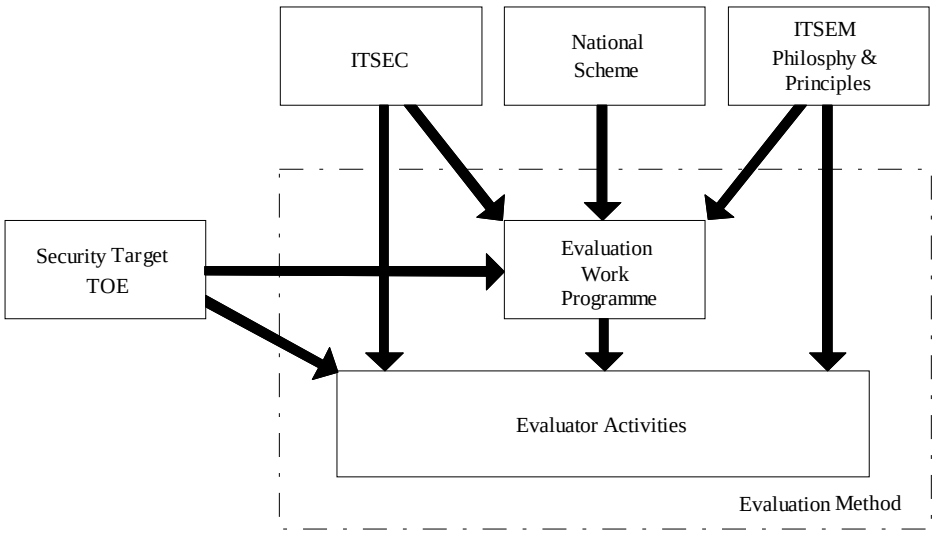


Figure 3.2.1 Derivation of the Evaluation Process

3.2.16 The instantiation of the evaluation method in a particular evaluation process is affected by:

a) evaluation attributes (concurrent or consecutive evaluation);

b) TOE attributes (system or product).

3.2.17 These attributes are described in part 1 of the ITSEM (1.2.24 - 1.2.28).

Chapter 3.3 Security and Evaluation Concepts

3.3.1 In this chapter a number of ITSEC concepts and terms are further explained, in addition to the concepts introduced in part 1 of the ITSEM, to guide the interpretation of certain evaluator activities. The concepts and terms are:

a) security objectives, **assets**, and threats;

b) correctness and effectiveness;

c) components, functions and mechanisms;

d) security enforcing, relevant and irrelevant functions and components;

e) separation of functionality;

f) refinement, **errors**, and error correction;

g) construction and **operational vulnerabilities**;

h) strength of mechanisms;

i) exploitable vulnerabilities;

j) penetration testing.

3.3.2 It should be noted that national schemes may provide further interpretation of these and other terms.

Security Objectives, Assets, and Threats

3.3.3 The security target specifies the security objectives of the TOE, relating threats and assets to each objective (there must be at least one security objective). An example objective might be:

a) The TOE shall prevent the disclosure of sensitive information to personnel with insufficient clearance to access that information.

b) The TOE shall ensure that supervisors charged with cross-checking of customers' data do not abuse their authority in order, for example, to commit fraud.

3.3.4 The security target enumerates the threats to the TOE and the assets that the TOE shall protect. The ITSEC requires threats and assets to be identified so that the evaluators can check that the security objectives and the individual lists of threats and assets are consistent with each other.

3.3.5 The security target also identifies the countermeasures that are to be implemented to protect the assets from the threats in order to satisfy the security objectives. Where the countermeasures are to be enforced by technical means, i.e. within the computer system itself, they are referred to as security enforcing functions. Such functions specify the security functionality of the TOE (rather than the mechanisms that will be used to implement security functions). The ITSEC recommends that such functions are described under the generic headings specified in Chapter 2 of the ITSEC, or by use of a predefined functionality class. The security target also identifies the particular objectives of each countermeasure, e.g. the TOE employs an identification and **authentication** function to establish and verify a claimed identity.

3.3.6 Specific threats and assets are more difficult to specify in the security target for a product than for a system. The product rationale may therefore be used by a purchaser to determine how his actual assets can be protected from his actual threats through the use of the countermeasures supplied in the product. The product rationale is therefore more likely to detail security objectives than known threats and assets.

Correctness and Effectiveness

3.3.7 Fundamental to the ITSEC criteria is the separation between functionality and assurance and the further split of assurance into confidence in the correctness of the security enforcing functions and confidence in the effectiveness of those functions.

3.3.8 Two key questions have to be answered during evaluation:

a) Do the deliverables demonstrate that the TOE correctly implements the security target? (correctness)

b) Are the security measures implemented in the TOE effective against the identified threats and free from exploitable vulnerabilities? (effectiveness)

3.3.9 Correctness is concerned with two main issues:

a) Is there an adequate description of the security enforcing functions in the security target, and do the deliverables provide evidence that these functions are correctly implemented in the TOE?

b) Has a disciplined development approach been followed, such that an adequate level of confidence in the **correct refinement** of the requirements can be established?

3.3.10 Effectiveness should be considered as a checklist including different aspects on which a TOE may fail. Effectiveness is concerned with the following questions:

a) Are the security enforcing functions able to protect the specified assets from the threats specified in the security target? (Suitability of Functionality)

b) Is the design such that, given the correct implementation of the individual security enforcing functions, the TOE as a whole will be secure when measured against its security target? (Binding of Functionality)

c) Does the TOE, as a whole and in its operational environment, have any exploitable vulnerabilities? (Vulnerability Assessments, Strength Of Mechanisms and Ease Of Use)

Components, Functions, and Mechanisms

3.3.11 The TOE is made up of components. Components themselves are made up of components, with the components identified by the developer at the lowest level of design being called basic components, e.g. compilation units.

3.3.12 A component may implement more than one function. In the case of a basic component those parts which contain the implementation of each such function are referred to as a functional unit. It is important that the security functions identified in the security target can be mapped to components on all levels of abstraction considered in the evaluation.

3.3.13 The logic or algorithm that implements a function is called a mechanism. Evaluation considerations concerning mechanisms are contained in part 6, annex 6.C.

Security Enforcing, Relevant and Irrelevant Functions and Components

3.3.14 The terms security enforcing, security relevant and security irrelevant are mutually exclusive, but all encompassing, i.e. every item of TOE functionality can be assigned to exactly one of these three categories. These three attributes can be applied to functions and components.

3.3.15 Functions are security irrelevant, if the fulfilment of the security objectives does not depend on them. Security enforcing functions are all functions of the TOE which directly contribute to the security objectives. Security relevant functions contribute to the secure functioning of the TOE and will often provide services not just to the security enforcing functions but also to non-security related functions. Usually, security enforcing functions rely on the correct operation of the security relevant functions.

3.3.16 If at least one of the functions implemented in a component is security enforcing, then that component is security enforcing. If none of its functions is security enforcing or relevant, then the component is security irrelevant.

Separation of Functionality

3.3.17 Separation may be shown by demonstrating (with appropriate rigour) that whatever behaviour the non-security enforcing components exhibit, provided that the security enforcing components function correctly, then the security objectives will be upheld.

3.3.18 The separation between security enforcing, security relevant and security irrelevant functions is an architectural issue not solely determined by security considerations. Through the reference monitor concept it is known how to separate functionality supporting confidentiality requirements. However, this concept cannot successfully be extended to cover the areas of integrity and availability.

Refinement, Errors and Error Correction

3.3.19 The ITSEC criteria do not prescribe any particular development method, but it is assumed that the development of any TOE involves several stages of refinement and integration. At the end of the development process there are representations of the TOE at different levels of abstraction. The security target is on the highest level of abstraction. The operational TOE in the form of executable code or electronic circuitry is the most concrete and detailed representation. In the ITSEC correctness criteria, the terms *security target*, *architectural design*, *detailed design* and *implementation* denote different levels of abstraction. The detailed design, for example, is less abstract and more detailed than the architectural design. Therefore the detailed design is called a refinement of the architectural design.

3.3.20 A function described in the security target occurs at different levels of abstraction or detail including its implementation in the TOE. The description of that function at any given abstraction level in this hierarchy is said to be a correct refinement if the totality of effects described at that (lower) level of abstraction exhibits the effects described at the preceding (higher) level of abstraction.

3.3.21 A failure to meet the correctness criteria is called an error. A typical cause is an inconsistency with respect to refinement. It could also be considered as a problem of traceability or an inconsistency between two representations of the TOE. The purpose of the ITSEC correctness criteria is to help establish the fact that each representation provided to the evaluators is a correct refinement of its corresponding higher level representation. The ITSEC correctness criteria for construction try to provide evidence that the TOE is a correct refinement of the security target. The mapping between the security target and the TOE is provided by mappings between intermediate levels as illustrated in figure 3.3.1.

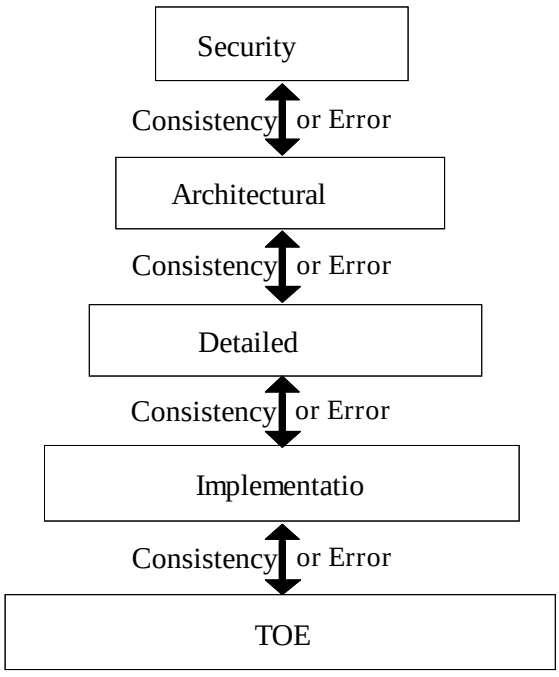


Figure 3.3.1 Representations of the TOE and Correctness

3.3.22 An error is corrected by modification of at least one of the representations. For example, suppose that at one level of design there is a representation of an identification and authentication function. The design will specify the action to be taken in the event of overflow of the tables holding the user identities and passwords. If at the next lower level of design a different action is taken in the event of overflow of the tables, it is an error since an effect specified on one level is not present on the next level. Therefore:

a) Either the design on the first level is modified to specify the action in fact taken on the next level. It may impact higher levels of design, e.g. the architectural design and perhaps the security target.

b) Or, the design on the next level is modified to specify the action as required on the first level. This usually affects levels of design and implementation below the next level.

3.3.23 Another typical cause of an error is the provision of insufficient evidence by the sponsor/developer. Undetected errors could lead to potential vulnerabilities. Typographical errors in the sponsor/developer documentation are usually not classed as an error in ITSEC terms.

Construction and Operational Vulnerabilities

3.3.24 A **vulnerability** is a security weakness in a TOE which can be used by an attacker to exert a threat and endanger an asset or defeat a countermeasure. There are construction and operational vulnerabilities. **Construction vulnerabilities** take advantage of some property of the TOE which was introduced during its construction, e.g. failure to clear a buffer. Operational vulnerabilities take advantage of weaknesses in non-technical countermeasures to violate the security of the TOE, e.g. disclosure of one's password to someone else.

3.3.25 Refinement often generates additional detail at the lower level of abstraction. The effects at the lower level are a "superset" of the effects at the higher level. The added details are sources of potential construction vulnerabilities. For example, the introduction of a lock variable which is not present at the higher abstraction level introduces a potential vulnerability. If information flow control is a security objective in the security target and the lock variable can be used to create a covert channel, the vulnerability may be exploitable.

3.3.26 Potential vulnerabilities from refinement are identified by the evaluators while performing the correctness assessment. The construction vulnerability assessment decides whether those vulnerabilities are exploitable or not.

3.3.27 Operational vulnerabilities concern the boundary between IT and non-IT countermeasures, e.g. operational procedures concerned with physical security, non-electronic forms of key management, the distribution of security badges. Non-IT measures will be of concern to the evaluators if:

a) they appear as part of the operational documentation, or

b) the security target is formulated on the basis of a System Security Policy (see ITSEC Paragraphs 2.8 - 2.15), or they appear as part of the Product Rationale (see ITSEC Paragraphs 2.16 - 2.17).

3.3.28 The non-IT countermeasures on which the security of the TOE depends are identified as assertions about the environment of the TOE. For example, it may be asserted that only company employees are allowed access to the system and that it is the responsibility of non-IT countermeasures to ensure that this assertion is satisfied. The evaluators assume that this assertion is true. If the combination of IT and non-IT countermeasures are in the scope of the evaluation, the evaluators should determine whether the combination contains any potential vulnerabilities.

Strength of Mechanisms

3.3.29 The ITSEC definitions for the three ratings of the strength of mechanisms *basic*, *medium* and *high* are a coarse scale for expressing user needs. The definitions do not provide a detailed means for assessment during evaluation. A distinction has to be made between the amount of effort needed to discover a vulnerability, to find a description of a vulnerability (e.g. reading about it in a magazine), and finally to exploit a vulnerability based on the description. The emphasis of the rating is on the amount of effort required to exploit a vulnerability.

3.3.30 The evaluator rating of the strength of mechanisms is based on the aspects of expertise, opportunity and resources. More practically the four parameters expertise, collusion, time, and equipment can be used.

3.3.31 The rating should be calculated for all possible and reasonable combinations of the values for the parameters. This can be accomplished by the use of tables or a set of rules. Details on the assessment of the strength of mechanisms are presented in part 6, annex 6.C.

3.3.32 Cryptographic mechanisms are not rated by ITSEFs (see ITSEC Paragraph 3.23).

Exploitable Vulnerabilities

3.3.33 There may be many ways to defeat a particular countermeasure, some ways being easier than others. Usually there is more than one countermeasure that an attacker must defeat in order to successfully attack the TOE. The developer anticipates the ways in which the TOE might be attacked and selects countermeasures accordingly. Based on the developer's analysis, the evaluators independently investigate the TOE from the viewpoint of an attacker to determine any way in which a security objective may be compromised.

3.3.34 A successful penetration reveals an exploitable vulnerability or a failure to meet the strength of mechanisms required. If there is a successful attack, the vulnerability is exploitable. In the interests of cost-effective evaluation, vulnerabilities do not have to be demonstrated exploitable by testing if theoretical arguments are enough. Attack scenarios are developed and penetration tests are performed as part of the vulnerability assessment in the evaluation.

Penetration Testing

3.3.35 When the evaluators have established the list of potential vulnerabilities and compared it with the list provided by the developer (ITSEC Paragraph 3.12), the evaluators complete the independent analysis by performing penetration tests to check whether the potential vulnerabilities are exploitable.

3.3.36 Penetration testing is different from functional testing, which attempts to demonstrate that the TOE conforms to its specification.

Chapter 3.4 Principles of the Conduct of Evaluations

Theory and Experiment

3.4.1 Theories about the TOE and its behaviour may help evaluators to understand how the TOE meets its security target. Evaluators should build and record their theories of the TOEs during the analysis of the deliverables. These theories should be adopted and confirmed, or rejected, by considering other information about the TOE, or experimentally by penetration and other tests.

3.4.2 In the field of science, experiment is directed by a hypothesis which is then tested. Such experiments can be classified as one of the following:

a) tests to show that the system under consideration does or does not have certain properties;

b) attempts to distinguish between competing theories about the system's behaviour by devising and performing experiments to verify or falsify the different theories.

3.4.3 This principle on experiments and theories can be applied to the practice of evaluation. Tests of TOEs should not be undertaken at random, but driven by a theory or suspicion to be checked. There are several ways in which evaluators can proceed. From the analysis of the security target the evaluators should come to an understanding of the security properties required of the TOE, and use this information to develop tests. From analysis of the other evaluation deliverables, the evaluators should understand the behaviour of the TOE, and use this information to develop tests to confirm or reject that potential vulnerabilities are exploitable. Another important source for developing tests is knowledge about the behaviour of similar products and systems.

Systematic Decomposition

3.4.4 The complexity of a TOE is practically unlimited. Systematic decomposition is a well known approach to cope with this problem during evaluation. This approach is reflected in various ITSEC requirements on the developer concerning evaluation deliverables and the development process. Examples are:

a) division of the statement of required security functionality into security enforcing functions in the security target;

b) architectural separation of security functionality from other functionality;

c) use of a phased construction process;

d) use of structured development approaches;

e) use of programming languages which encourage modularisation.

3.4.5 The ITSEC criteria also follow the principle of systematic decomposition during evaluation by separating the aspects of correctness from effectiveness and distinguishing different aspects of effectiveness like suitability, binding etc.

Modelling

3.4.6 Modelling is used as an evaluation technique to support theory and demonstrate understanding. It is particularly relevant for higher evaluation levels. Development of models is often based on experience and intuition. They are described using an informal, semiformal, or formal specification style. Models provided by the sponsor/developer should be used by evaluators as a base for their understanding and modelling.

Traceability

3.4.7 For higher evaluation levels the fulfilment of the security objectives should be completely traceable down to the operational TOE by the evaluators. This traceability can only be complete if it covers all development phases. This includes the requirements, architectural design, detailed design and implementation phases. The traceability has to be provided by the security target and the other deliverables which provide different representations of the TOE. This also covers source code and executable code if applicable for the evaluation level and the TOE in question.

Verdicts

3.4.8 With regard to the ITSEC criteria a TOE successfully passes the evaluation only if it achieves pass verdicts for all the correctness and effectiveness criteria for the targeted evaluation level. This implies that in the conclusion phase of the evaluation no exploitable vulnerabilities remain in the operational TOE, and the claimed minimum strength of mechanisms has been met. The evaluation fails if in the end at least one of the correctness criteria is not fulfilled or if an exploitable vulnerability remains in the TOE.

3.4.9 The starting point for the evaluators to assign a verdict to an ITSEC criterion is the evidence the sponsor provides in the deliverables. It is supplemented by additional evaluator actions according to ITSEC, usually by a cross check or some kind of penetration testing, to provide independent evidence for the satisfaction of the criterion and to check the validity of the sponsor/developer evidence. This principle of independence applies to all results of sponsor/developer analysis and tests, for example, confirming test results by repeating samples. A fail verdict is assigned if no evidence, incomplete evidence (principle of completeness), or incorrect evidence for a relevant criterion is supplied by the sponsor/developer.

Error Correction

3.4.10 If an error is detected during the evaluation, it is necessary to fix it or the evaluation will eventually return a fail verdict on one of the correctness criteria. The same holds for exploitable vulnerabilities.

3.4.11 Corrections to previously evaluated deliverables will invalidate some of the previous evaluation work, necessitating repetition of evaluation work.

Penetration Testing

3.4.12 Penetration testing provides independent assurance that a specific TOE does not contain exploitable vulnerabilities, or critical mechanisms with a strength of mechanisms lower than claimed.

3.4.13 Penetration testing is the culmination of the following process:

a) gaining an understanding of the TOE and the security target during the correctness evaluator actions;

b) searching for vulnerabilities and generating hypotheses about their exploitation during the effectiveness evaluator actions.

3.4.14 Penetration testing is performed for all evaluations and is usually the final evaluator activity. Evaluators identify, specify, perform, and record penetration tests.

Checklists

3.4.15 Checklists used in evaluations can ensure that no relevant standard issue, e.g. well-known vulnerabilities in a certain type of product or system, is forgotten before passing a verdict.

Review

3.4.16 In evaluations, thought and judgement are required. To limit bias and the consequences of mistakes and to ensure the overall quality, the results of evaluation activities should be subject to a review process within the ITSEF. Requirements concerning the review process and the involvement of the **certification body** may be detailed in the national scheme. The review should involve at least one person who has not participated in the production of the result being reviewed.

3.4.17 The purpose of the evaluation review process is to ensure that the results of the evaluation are in accordance with the relevant criteria, the requirements of the ITSEM and the national scheme.

Records

3.4.18 Extensive record-keeping is required to provide evidence of the evaluation work and results. Important decisions, arguments, tests, and their results should be documented e.g. in log books or reports. Documentation of temporary problems and their solution or actions independently performed by the evaluators could be considered as useful and corroborative. The rules of the national evaluation and certification scheme may apply to this aspect.

Resources

3.4.19 Resources required for an evaluation mainly depend on the complexity of the TOE, its security target and the evaluation level. Other factors affecting the necessary amount of resources are the competence and experience of evaluators and the use of support tools. The necessary evaluator actions are derived from the relevant set of criteria, the structure of the TOE, and the evaluation deliverables. Efficiency is a concern of the ITSEF. Minimum requirements for resources are a matter of the national scheme and should be based on practical experience.

Resources for Penetration Testing

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.

Evaluation Work Programme

3.4.21 An evaluation work programme details the evaluator activities, estimates the necessary resources, and establishes a time scale. Guidance for writing an evaluation work programme is given in part 4 of the ITSEM.

Repeatability, Reproducibility, Impartiality, and Objectivity

3.4.22 Repeatability, reproducibility, impartiality, and objectivity are principles worth striving for in performing evaluations. They are closely related to each other, in particular impartiality to objectivity and reproducibility to repeatability. Impartiality and objectivity are prerequisites for reproducibility and repeatability. This is illustrated in figure 3.4.1.

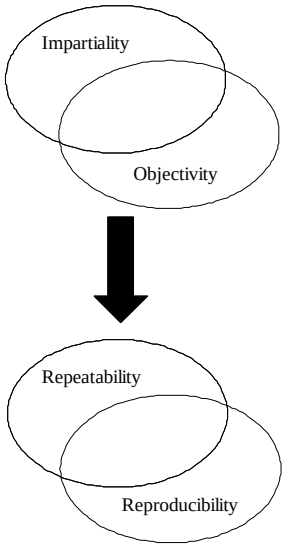


Figure 3.4.1 Four Basic Principles in Evaluation

3.4.23 The use of standardised documented evaluation procedures, techniques, and tools can support the four basic principles (see part 4 of the ITSEM). The aspects of effectiveness such as identification of vulnerabilities, strength of mechanisms, and exploitability of vulnerabilities are a special concern, as they introduce subjective factors like experience and intuition. Subjectivity cannot completely be countered within the evaluation process. It needs the involvement of a party with independent oversight of evaluations such as a certification body, which will ensure consistency and comparability between the results of different ITSEFs (see part 2 of the ITSEM).

Part 4 Evaluation Process

Contents

Figures

Chapter 4.1 Introduction

Evaluation Methods

4.1.1 This part of the ITSEM is addressed specifically to evaluators. The methods used in evaluations are described, both in terms of the organisational framework and in terms of the techniques used to evaluate TOEs against the ITSEC. The evaluation inputs, process and outputs are also described. It does not give an exhaustive description of how every evaluator action is performed.

4.1.2 Certain sections of this part of the ITSEM define aspects of evaluation methods which are mandatory - these sections are clearly identified in the text by using bold and grey shading. The objective of this prescriptive material is to ensure that evaluations performed using ITSEC and ITSEM have a common technical basis.

Structure

4.1.3 The structure of this part is a set of chapters, with these introductory remarks forming chapter 4.1.

4.1.4 Chapter 4.2 provides an overview of the evaluation process, with an identification of the roles of those involved in evaluation and a description of the phases through which the process passes.

4.1.5 Chapter 4.3 describes the arrangements for initiating evaluation and obtaining **deliverables**.

4.1.6 Chapter 4.4 contains a detailed description of the process of evaluation from the point of view of the evaluators. The level of detail given is that which is necessary to provide technical equivalence of evaluations.

4.1.7 Chapter 4.5 discusses techniques and tools of use to evaluators.

4.1.8 Chapter 4.6 advises evaluators on re-using evaluation results.

4.1.9 Chapter 4.7 specifies the outputs that an evaluation should produce, that is, **Evaluation Technical Reports** (ETRs).

Chapter 4.2 The Evaluation Process

Introduction

4.2.1 This chapter provides an overview of the evaluation process, defining the roles of those involved in the process, and the phases and stages that the process goes through.

4.2.2 The evaluation process described in this chapter is to be seen as a framework which describes organisational and procedural aspects to be followed during the conduct of an evaluation.

Roles

Overview

4.2.3 The evaluation process described in this chapter requires the existence of the following bodies:

a) ITSEF;

b) sponsor;

c). developer;

d) **certification body**.

4.2.4 The role of each of these bodies in the evaluation process is given in the following subsections. Figure 4.2.1 shows these bodies and the types of information which might flow between them during the evaluation process.

ITSEF

4.2.5 The role of the ITSEF is to act as an independent body in which third party evaluations can be performed within the framework of the **national scheme**. The ITSEF supports the organisational, administration and contractual aspects of evaluations.

4.2.6 Within the ITSEF, it is the role of the evaluators to perform a detailed impartial examination of a TOE to search for **vulnerabilities** and to determine the extent to which its security target is met by its implementation, in accordance with the ITSEC. The results of the evaluation are supplied to the certification body and to the sponsor.

4.2.7 The evaluators perform the evaluation work in accordance with the ITSEC/ITSEM requirements and the practices and procedures laid down by the national scheme. In performing this work, the evaluators shall be responsible for:

a) maintaining a record of all the work performed during the evaluation;

b) production of evaluation reports;

c) maintaining confidentiality as necessary on all aspects of the evaluation work.

4.2.8 Evaluators provide support to the certification body during the certification process (see the *Phase III Conclusion* subsection in this chapter).

4.2.9 Evaluators liaise with other parties involved in the evaluation, which will include the sponsor of the evaluation, the developer of the TOE and the certification body.

4.2.10 The evaluators should ensure that the sponsor and developer(s) are aware of, and fully understand, obligations placed upon them by the national scheme. In particular, the evaluators should ensure that the sponsor is able to supply all the necessary inputs to the evaluation process (deliverables). Therefore, at the start of an evaluation the evaluators should clearly identify what the sponsor must provide.

Sponsor

4.2.11 The sponsor of an evaluation will typically be the Vendor of a product, or the User or Supplier of a system, wishing to demonstrate that the TOE meets the specified security target.

4.2.12 The sponsor initiates the evaluation of a TOE by an ITSEF. He defines the security target, commissions the evaluation, receives the ETR and, if the evaluation returns a pass verdict, the **certificate/certification report**.

4.2.13 The role of the sponsor is described in more detail in part 6 of the ITSEM.

Developer

4.2.14 The term developer is used to refer to the organisation (or organisations) which produces the TOE (or component parts of the TOE). The developer (if not also sponsoring the evaluation) should be prepared to cooperate with the sponsor and agree to assist in the evaluation, e.g. by providing technical support to the ITSEF.

4.2.15 The role of the developer is described in more detail in part 6 of the ITSEM.

Certification Body

4.2.16 The main aims of a certification body are:

a) to create the conditions under which the work of all the ITSEFs in a scheme will be accurate and consistent and their conclusions valid, repeatable and reproducible;

b) to provide independent confirmation that evaluations have been carried out in accordance with approved criteria, methods and procedures.

4.2.17 The role of the certification body is described in more detail in part 2 of the ITSEM.

Phases of the Evaluation Process

Overview

4.2.18 The evaluation process can be divided into three phases: preparation, conduct and conclusion. The three phases are described in detail in the following Subsections.

Phase I - Preparation

4.2.19 The sponsor approaches the relevant body under the national scheme (certification body or ITSEF) and initiates the evaluation of a TOE. An ITSEF to be contracted by the sponsor for Phase I is selected. The sponsor supplies the ITSEF with his security target for the TOE (possibly in draft form), and defines the scope of the evaluation.

4.2.20 The ITSEF assesses the likelihood of a successful evaluation, requesting relevant information from the sponsor. If it is satisfied, it will agree a contract with the sponsor to perform the evaluation. Optionally, the ITSEF may review the security target and advise the sponsor about changes necessary to ensure a firm basis for the evaluation.

4.2.21 National schemes may require ITSEFs to prepare an **Evaluation Work Programme** (EWP) or a deliverables list before starting the evaluation. An EWP is a description of the work the ITSEF will do during the evaluation. A deliverables list is a description of what the sponsor will be expected to provide to the ITSEF during the evaluation, including the dates when they will be required. The certification body will review the EWP to ensure that the work proposed is adequate. The advantage of EWPs and deliverables lists is that the sponsor and ITSEF can be clear from the outset about what work is required.

4.2.22 At the start of an evaluation, an EWP will be based on the information available to the ITSEF at that time. As the evaluation progresses, the information available to the evaluators can be expected to increase, and the EWP will evolve. The certification body will review the modifications of the EWP to ensure that the work proposed is adequate.

4.2.23 ITSEFs can provide advice to sponsors and developers about how to produce the necessary evaluation deliverables. Advice can be given by a different ITSEF than that which performs the evaluation. **If an ITSEF gives advice, the advice given must not be such as to affect the ITSEF's independence in any evaluation**. Details will be given by national schemes.

4.2.24 During the preparation phase, the sponsor and ITSEF should agree on the necessity for re-evaluation information in the ETR.

Phase II - Conduct

4.2.25 By the Conduct phase, a contract should have been agreed between the sponsor and ITSEF, the work required should be understood, and the security target should be stable.

4.2.26 For each relevant phase or aspect of the ITSEC, the evaluators perform the required evaluator actions. The deliverables are checked to see if every criterion is addressed. In addition the evaluators produce a list of **potential vulnerabilities**. All problems identified will be discussed between the appropriate parties.

4.2.27 Problems identified in the conduct phase fall into two different groups. The first group represents problems where the sponsor is able to provide a solution which is acceptable to the ITSEF and certification body. A period of time is agreed between ITSEF and sponsor within which the problem is solved according to the agreed solution. The second group represents problems where the sponsor is unable or unwilling to fix them. The ITSEF and certification body inform the sponsor about the implications if the problem is not fixed. The sponsor may then either abandon the evaluation, or accept the implications for certification.

4.2.28 **During an evaluation, the ITSEF must produce its ETR**. The ETR is the final product of the evaluation but it does not represent the final product of the evaluation and certification process. The final draft of the ETR is distributed to the sponsor and to the certification body for approval.

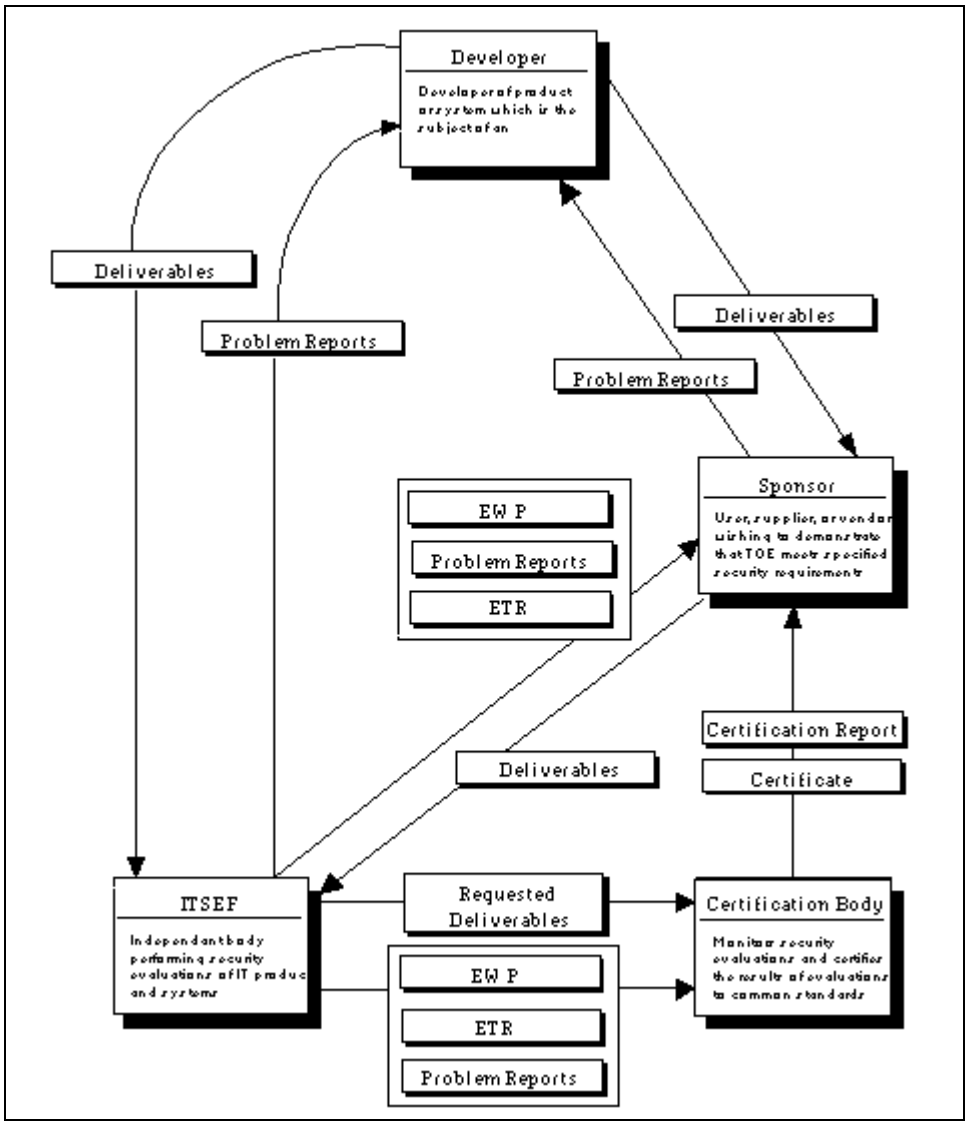
Phase III - Conclusion

4.2.29 In the Conclusion Phase the ITSEF provides the accepted version of the ETR to the certification body and the sponsor as a record of the results of the evaluation. The ETR should also be useful to future evaluators if the TOE is re-evaluated. The ETR, or parts of it, whether in draft or in final form, should be treated in confidence.

4.2.30 The certification body may request the ITSEF to provide technical support, and may make reasonable requests for access to specific technical evidence and results, to support conclusions presented in the ETR. This will not normally involve the ITSEF in additional evaluation work.

4.2.31 During the certification process, the certification body reviews the ETR to determine whether the security target is met by the TOE, taking account of any factors outside the scope of the evaluation. As part of the process it is able to assign an evaluation level. Its conclusions are recorded in the certificate/certification report.

4.2.32 Disposal of deliverables must be done in this phase.



Note: A sponsor may also be a developer

Figure 4.2.1 Example Information Flow in the Evaluation Process

Chapter 4.3 Inputs to Evaluation

Introduction

4.3.1 This chapter describes factors that evaluators should take into account before and during the initiation of an evaluation. As such it is concerned with how evaluators assist in the provision of inputs to the evaluation by the sponsor and what evaluators do with the deliverables, in terms of their handling, once they have been received.

4.3.2 The purpose of this chapter is not to mandate how national schemes are organised, but rather to provide evaluators with information on how an evaluation will typically be initiated and the evaluation deliverables handled.

4.3.3 It should be noted that whilst initiation of an evaluation is not mandated to be through the auspices of the certification body, it is recommended that the certification body be involved in an evaluation as early as is reasonable in order that the technical and commercial risks to the evaluation are minimised.

4.3.4 The term *deliverable* is used to refer to any item (including the TOE itself) that is required to be made available to the evaluators for evaluation purposes. This includes intangible items, such as support to the evaluators and access to computers. Part 6, annex 6.A should be consulted for details of the deliverable requirements of the ITSEM/ITSEC.

4.3.5 The purpose of deliverables is to allow the evaluators to evaluate the TOE. Different types of deliverables satisfy this purpose in different ways, e.g.:

a) Deliverables may provide evidence of effectiveness or correctness, e.g. an informal description of correspondence between source code and detailed design.

b) Deliverables may enable the evaluators to establish additional evidence of effectiveness or correctness, e.g. access to the developed TOE.

c) Deliverables may improve the overall efficiency of the evaluators' work, e.g. technical support from the developer.

Responsibility for Deliverables

4.3.6 The responsibility to provide all the required deliverables for an evaluation lies with the sponsor. However, most of the deliverables will be produced and supplied by the developer (where the sponsor is not the developer). The evaluators are not concerned with the contractual relationship between the sponsor and developer. The sponsor is the evaluators' client.

4.3.7 The running costs and risks (e.g. loss or damage through fire, flood, theft, etc) in all deliverables should be the responsibility of the sponsor, unless specifically agreed otherwise with the evaluators. It should be noted that some deliverables, such as new or special purpose types of hardware may not have an easily identified replacement cost and may well present insurance risks that cannot be transferred to evaluators.

4.3.8 It is recommended that the evaluators produce a deliverables list. This is a definitive list of the expected evaluation deliverables (for instance as a set of references to the sponsor's documentation) with the dates at which the evaluators expect the deliverables to be available to them. The deliverables list can be referenced in the ETR.

4.3.9 It is recommended that the objectives for initiation of the evaluation are clearly understood by the evaluators and promulgated to other parties. Therefore, it is recommended that evaluators ensure that all parties involved in the evaluation have a common understanding of the purpose and scope of the evaluation and are aware of their responsibilities.

4.3.10 Examples of issues that may need to be covered with a sponsor include national sensitivity and commercial confidentiality of information, access to or requirements for specialist tools, any limitations imposed on the access of evaluators to the evaluation deliverables, any previous evaluation results and the desired frequency of progress meetings.

4.3.11 For a particular arrangement between an ITSEF and a sponsor, the following details may have to be clarified:

a) the medium and format of computer-readable deliverables;

b) the schedule for the deliverables' production;

c) the number of copies of deliverables to be supplied;

d) the position regarding draft deliverables;

e) the position regarding any products to be used in conjunction with the TOE;

f) arrangements for discussing the development environment with the developer;

g) access to the operational and development sites;

h) type and duration of developer support, including computer access and requirements for office accommodation for evaluators.

4.3.12 In many cases the evaluators will require access to information provided by subcontractors, or third parties. The sponsor should take such cases into account.

4.3.13 The issue of whether the evaluation is concurrent or consecutive will impact the availability of deliverables and will need to be taken into account while producing a specific EWP (see chapter 4.4).

4.3.14 The issue of whether the evaluation is of a system or a product will also impact the provision of evaluation deliverables and hence a specific EWP. For instance, a product may be available for installation and testing at the ITSEF, whereas a system is unlikely to be made available to the ITSEF in this way.

Management of Deliverables

Confidentiality

4.3.15 During their work, ITSEFs will be given access to their clients' commercially sensitive information, and may gain access to nationally sensitive information. Evaluation partners must have confidence that the information given to ITSEFs will not be misused.

4.3.16 General requirements for confidentiality are a matter for national schemes. Sponsors and ITSEFs may agree additional requirements as long as these are consistent with the national scheme.

4.3.17 Confidentiality requirements will affect many aspects of evaluation work, including the receipt, handling, storage and disposal of deliverables.

Draft Deliverables

4.3.18 Evaluators require stable and formally issued versions of deliverables. There may be occasions, however, when it is helpful for the evaluators to also see draft versions of particular deliverables, e.g.:

a) test documentation, to allow the evaluators to make an early assessment of tests and test procedures;

b) source code or hardware drawings, to allow the evaluators to assess the application of the developer's standards.

4.3.19 Draft deliverables are more likely to be encountered where the evaluation of a TOE is performed concurrently with its development. However, they may also be encountered during the consecutive evaluation of a product or system where the developer has had to perform additional work to address a problem identified by the evaluators (e.g. to correct an error in construction) or to provide evidence of security which is not provided in the existing documentation (e.g. effectiveness deliverables in the case of a product or system not originally developed to meet the requirements of the ITSEC).

Configuration Control

4.3.20 **Evaluators must exercise control over evaluation deliverables so that the certification body can assure that the evaluation results are relevant to the (ultimately) operational TOE**. The evaluators should operate a quality system that seeks conformance with [EN45] such that the evaluation deliverables can be controlled and managed in accordance with the sponsor's wishes and the national scheme.

Disposal of Evaluation Deliverables

4.3.21 Evaluators must dispose of all evaluation deliverables when the evaluation has been completed (at the end of the Conclusion phase). This may be achieved by one or more of:

a) destroying the deliverables;

b) returning the deliverables;

c) archiving the deliverables.

4.3.22 Any archived material must be kept in accordance with the requirements of the national scheme.

4.3.23 The rules for the disposal of the evaluation deliverables should be agreed with the sponsor before the beginning of phase II (Conduct).

Re-evaluation and Re-use Deliverables

Overview

4.3.24 This section provides guidance to evaluators on the inputs required for evaluations where previous evaluation results are part of the input available. Chapter 4.6 describes how these evaluations are performed.

4.3.25 **Re-evaluation** of a TOE may be performed when the TOE or its associated evaluation deliverables change. Examples of changes include increasing a TOE's target evaluation level or adding security enforcing functions to a TOE's security target. The sponsor performs an impact assessment and determines the appropriate course of action in order that the previous evaluation results can be re-affirmed according to the guidance contained in part 6, annex 6.D.

4.3.26 **Re-use** of evaluation results is a technique for reducing the evaluation effort for an evaluation of a TOE which includes one or more previously evaluated TOEs. The results of the original evaluation may or may not be valid in the context of the evaluation of the new TOE.

4.3.27 If the evaluation level(s) of the previously evaluated component(s) is greater than, or equal to, the target evaluation level of the TOE, then the previous correctness results are confirmed by the certificates/certification reports.

4.3.28 When a certified product or system is used as a component of a new TOE, the context of its use will have changed. Hence, whilst the correctness of the certified component with respect to its original security target is still valid, its effectiveness with respect to its new security target in the context of its new use needs to be re-affirmed.

4.3.29 The sponsor will therefore be expected to provide the new TOE deliverables for the target evaluation level together with the certificates/certification reports for any certified components.

4.3.30 However, the correctness deliverables for the TOE's components may be required to support the effectiveness analysis.

4.3.31 The new TOE deliverables for effectiveness will need to cover the effectiveness of the pre-evaluated product(s) as they are used in their new context. For instance, the security target for the new TOE will have to be demonstrated to make suitable use of the pre-evaluated product(s). Similarly, the binding between *all* the new TOE components will need to be addressed by the sponsor, even when some of these components have certificates/certification reports resulting from their original evaluation(s).

Availability of Evaluation and Certification Results

4.3.32 The certificate/certification report or the ETR (or parts of it) might be input to re-evaluation or re-use. In practice, the extent to which the results of previous evaluations are available, depends on whether they have been performed:

a) by the same ITSEF;

b) by a different ITSEF under the terms of the same national scheme, and;

c) by an ITSEF under the terms of a different national scheme.

4.3.33 The ETR may contain commercially and nationally sensitive information that should not be disseminated to a wide audience. Hence, the ETR can only be guaranteed to be available to ITSEFs operating within a particular national scheme. Also, re-evaluation information is optional and may not always be present in the ETR.

4.3.34 The certificate/certification report will be publicly available and will provide a summary of the TOE and its evaluation (see part 2 of the ITSEM). Hence, sponsors will always be able to provide certificates/certification reports to ITSEFs.

4.3.35 For the availability of evaluation results in all cases the rules of the national schemes will apply..

Chapter 4.4 Conduct of the Evaluation

Introduction

4.4.1 This chapter establishes the process of evaluation that is recommended for all evaluations against the ITSEC, and lays down some mandatory requirements. The procedural aspects of organising this process (for instance the detailed procedures to be followed for problem **reporting** throughout evaluations) are not prescribed in this chapter. These are left to the discretion of national schemes.

4.4.2 The process of evaluation is designed to conform to the philosophy and principles established in part 3 of the ITSEM. The process concerns planning, performing and reporting the evaluation in such a way that it is easily shown to conform to the ITSEC and ITSEM.

4.4.3 The technical work performed is discussed in the remaining sections entitled *Work Programmes*, *Application of ITSEC* and in chapter 4.5.

Work Programmes

Overview

4.4.4 In order for a TOE to be certified, its evaluation must be conducted in conformance with the ITSEC/ITSEM and national scheme requirements.

4.4.5 An evaluation is carried out by performing each of the actions specified by the ITSEC. In order to describe the structure of an evaluation and the dependencies between evaluator actions this section introduces the concept of a generic EWP.

4.4.6 A generic EWP is a description of how the work required for evaluations is organised; that is, it is a description of the activities involved in the evaluation and the relationships between them.

4.4.7 A generic EWP is designed to be applicable to the evaluation of a wide range of systems and products. It is also designed to be broadly applicable to all the evaluation levels. Many generally applicable EWPs are conceivable; some of these will be more efficient and more flexible than others, but all may, potentially, implement valid interpretations of the ITSEC.

4.4.8 The generic EWP provides a simple expression of how an evaluation should be carried out in accordance with the evaluation philosophy and principles of part 3 of the ITSEM.

4.4.9 The required deliverables have been listed in part 6, annex 6.A. This section deals with the evaluation of those deliverables. Evaluation to ITSEC criteria involves:

a) checking that the deliverables conform to the ITSEC requirements;

b) checking that the security requirements specified in the security target are implemented adequately;

c) checking that no **exploitable vulnerabilities** exist in the operational TOE.

4.4.10 The above can be summarised as "Do All Correctness And Effectiveness Actions specified in the ITSEC". However, it is impossible to discuss each evaluator action at such a generic level as a diverse range of TOEs may need to be evaluated to any of the evaluation levels. Hence the concept of an activity is introduced such that the process of evaluation can be discussed generically.

4.4.11 It is important to be aware of the distinction between *actions* and *activities*. An *action* is an ITSEC evaluator action. An *activity* is a generic group of actions with some specific purpose, such as assigning a verdict to a particular **representation**.

Generic Evaluation Activities

4.4.12 The following (unordered) list identifies the names of the generic evaluation activities which will be performed during the evaluation phase. ITSEC paragraph numbers are shown in curly brackets {}; *n* below can be any number in the range 1 to 6:

Check **Suitability Analysis**	{3.16}
Check **Binding Analysis**	{3.20}
Examine Strength Of Mechanisms	{3.24}
Examine **Construction Vulnerabilities**	{3.28}
Examine Ease Of Use	{3.33}
Examine **Operational Vulnerabilities**	{3.37}
Check The Requirements	{En.4}
Check The Architectural Design	{En.7}
Check The Detailed Design	{En.10}
Check The Implementation	{En.13}
Check The Development Environment	{En.17, En.20, En.23}
Check The Operational Documentation	{En.27,En.30}
Check The Operational Environment	{En.34,En.37}
Perform Penetration Testing	{3.24, 3.28, 3.33, 3.37}
Write Reports	{5.11}

4.4.13 With the exception of *Perform Penetration Testing*, the technical evaluation activities correspond to the application of effectiveness or correctness criteria, as they appear in the ITSEC.

4.4.14 Regarding activity names, the only distinction between *Check* and *Examine* is that *check* principally involves analysing deliverables whereas *examine* also provides input into penetration testing. Penetration testing is explicitly related to these activities but has been assigned to a separate activity for two reasons:

a) to highlight that the previous analyses are consolidated and the tests devised during this activity;

b) to indicate that large sets of actual tests are normally carried out together.

4.4.15 The activity *Check Suitability Analysis* requires evaluators to check the developer's suitability analysis. This may expose vulnerabilities arising because a security enforcing function fails to uphold a security objective for a threat identified in the security target.

4.4.16 The *Check Binding Analysis* activity requires evaluators to examine the developer's binding analysis and establish whether or not the set of security enforcing functions, taken together, will adequately address all of the security objectives.

4.4.17 The *Examine Strength Of Mechanisms* activity requires evaluators to identify mechanisms which do not achieve the minimum strength of mechanisms required by the security target. Strength of mechanisms is discussed in part 6, annex 6.C.

4.4.18 During the correctness evaluation, the evaluators build up an understanding which is used during the *Examine Construction Vulnerabilities* activity to attempt to identify vulnerabilities in the construction of the TOE.

4.4.19 Errors found during correctness evaluation are one source of construction vulnerabilities. However, it is possible for a component to be regarded as correct (in the sense of **correct refinement**), but still to contain vulnerabilities. This is because:

a) as the refinement process progresses, new functionality is added;

b) the standard refinement verification techniques will not detect certain vulnerabilities such as covert channels.

4.4.20 This activity therefore requires evaluators to examine refinement errors, and additional functionality introduced at each development phase, in the search for exploitable vulnerabilities.

4.4.21 The *Examine Ease Of Use* activity requires evaluators to examine the insecure modes of operation of the TOE. Consequently, this assessment is closely related to the other operational assessments.

4.4.22 The *Examine Operational Vulnerabilities* activity requires evaluators to examine the operation of the TOE. The evaluators attempt to identify vulnerabilities in the way in which the TOE is operated.

4.4.23 Operational vulnerabilities concern the boundary between IT and non-IT **countermeasures**, such as operational procedures concerned with physical security, non-electronic forms of key management and the distribution of security badges. Non-IT countermeasures will be of concern to the evaluators if any of the following hold:

a) they appear as part of the operational documentation;

b) the security target is formulated on the basis of a system security policy (see ITSEC Paragraphs 2.8-2.15);

a) they appear as part of the product rationale.

4.4.24 Non-IT countermeasures generally arise when constructional vulnerabilities necessitate non-IT countermeasures to preserve the security of the TOE. Therefore, when evaluating operational vulnerabilities, evaluators are mainly concerned with ensuring that the non-IT countermeasures do counter the known constructional vulnerabilities.

4.4.25 The activity *Check The Requirements* requires evaluators to ensure that the security target adequately defines the security enforcing functions and is not self-contradictory. The security target should clearly identify the security enforcing functions, target evaluation level, strength of mechanisms rating and external security measures to be considered during the evaluation.

4.4.26 The first development step, from requirements to architectural design, is of particular importance as it provides the top level assignment of abstract functions to logical and physical components. An important assessment task for the evaluators, performed under the *Check The Architectural Design* activity, is to decide whether the separation of security enforcing from non-security enforcing functionality is `clear and effective', as this *permits evaluation effort to be concentrated on limited areas of the TOE that contribute to security, and enables the implementation of the security target to be easily followed, as the design is refined into further and further detail*. (Taken from Paragraph 4.20 of the ITSEC.)

4.4.27 The *Check The Detailed Design* activity requires evaluators to ensure that the separation policy is followed and that the security enforcing components have been correctly implemented. There may be several levels of detailed design.

4.4.28 Very similar remarks to the last paragraph apply to the implementation assessment using the activity *Check The Implementation*. The difference is simply level of detail, implementation dealing, by definition, with the further elaboration of the basic components and functional units identified by the last stages of detailed design (and hence functional testing becomes possible).

4.4.29 The activity *Check The Development Environment* requires the evaluators to check the development standards, particularly for languages to be used at various stages in the development. The evaluators must have confidence that the evaluated TOE matches the developed TOE, and that the notations used in implementation are unambiguous. This activity therefore addresses:

a) configuration control;

b) programming languages and compilers;

c) developer's security.

4.4.30 The *Check The Operational Documentation* activity requires the evaluators to check that the TOE can be administered and used in accordance with its security objectives.

4.4.31 The *Check The Operational Environment* activity requires the evaluators to check the TOE's correct delivery; showing that the operational TOE is a faithful copy of the development environment TOE and that it can be generated and operated in accordance with its security objectives.

4.4.32 The *Perform Penetration Tests* activity requires the evaluators to consolidate the `perform penetration tests' actions of the ITSEC (for instance those carried out during the *Examine Construction Vulnerabilities* activity), always for aspects of the effectiveness assessment, and always for the same purpose: to determine whether potential vulnerabilities can be exploited in practice.

4.4.33 The results of the evaluation have to be recorded so the activity *Write Reports* has to be introduced. The evaluators produce an ETR that corresponds to the requirements of chapter 4.7.

4.4.34 Figure 4.4.1 presents the actions of the ITSEC together with the activity that they relate to. In the figure, *check* * includes all the relevant checking actions which evaluators must perform according to the ITSEC. The other actions are presented in full.

Figure 4.4.1 Activities And Their Associated ITSEC Evaluator Actions

Activity	Action
Check Suitability Analysis	Check *
Check Binding Analysis	Check *
Examine Strength Of Mechanisms	Check *
Examine Construction Vulnerabilities	Check *, Perform an independent vulnerability analysis, taking into account both the listed and any other known construction vulnerabilities found during evaluation
Examine Ease Of Use	Check *, Repeat any configuration and installation procedure to check that the TOE can be configured and used securely, using only the user and administration documentation for guidance
Examine Operational Vulnerabilities	Check *, Perform an independent vulnerability analysis, taking into account both the listed and any other known operational vulnerabilities found during evaluation
Check The Requirements	Check *
Check The Architectural Design	Check *
Check The Detailed Design	Check *
Check The Implementation	Check * Use the library of test programs to check by sampling the results of tests Perform additional tests to search for errors Investigate any suspected inconsistencies between source code and executable code found during testing using the sponsor supplied tools
Check The Development Environment Configuration Control Prog. Languages And Compilers Developer's Security	Check *, Use the developers tools to create selected parts of the TOE and compare with the submitted version of the TOE Check * Check * Search for errors in the procedures
Check The Operational Documentation	Check *
Check The Operational Environment Delivery And Configuration Start-up And Operation	Check * Search for errors in the system generation procedures Check * Search for errors in the procedures
Perform Penetration Testing	(Strength of Mechanisms) Perform penetration testing where necessary to confirm or disprove the claimed minimum strength of mechanisms (Construction Vulnerabilities) Perform penetration testing to confirm or disprove whether the known vulnerabilities are actually exploitable in practice (Ease of Use) Perform other testing where necessary to confirm or disprove the ease of use analysis (Operational Vulnerabilities) Perform penetration testing to confirm or disprove whether the known vulnerabilities are actually exploitable in practice

Generic Evaluation Work Programme

4.4.35 The ITSEC introduces an implicit ordering of the activities by, for instance, stating that correctness and effectiveness are interleaved. However, this ordering needs to be made explicit.

4.4.36 The concept of an intermediate result is introduced to represent any information that is generated by the evaluators during one activity and used during another. It may be derived or copied from deliverables but is used solely by evaluators to perform their functions. These intermediate results should be recorded to aid **repeatability** and future re-evaluation. Intermediate results can be used to derive generic dependencies between evaluation activities and therefore to derive a schedule for the relevant activities.

4.4.37 Some intermediate results are directly derived from deliverables. These are:

a) the threats identified in the security target;

b) the external security measures identified in the security target;

c) the security enforcing functions identified in the security target;

d) the security relevant events identified from the requirements;

e) the components in the architectural design (and their types - security enforcing, security relevant or other);

f) the security relevant functions identified in the design;

g) the security mechanisms identified in the deliverables;

h) the security administration functions identified in the operational documentation;

i) the critical security mechanisms identified in the strength of mechanisms analysis.

4.4.38 Other intermediate results arise from the extra work done by evaluators:

a) the potential construction vulnerabilities identified by the evaluators;

b) the potential operational vulnerabilities identified by the evaluators;

c) the errors identified by the evaluators;

d) the penetration tests defined by the evaluators;

e) the exploitable vulnerabilities identified by the evaluators.

4.4.39 Finally, the ETR produced during the evaluation is a result.

4.4.40 Figure 4.4.2 presents a table of the evaluation activities and evaluation products. The evaluation products are either output from an activity (represented by `O') or input to an activity (represented by `I') for use during that activity. The output from an activity may be either a complete product or a contribution to an evaluation product produced by another activity. For instance several effectiveness activities contribute to the construction vulnerability list produced by the evaluators which subsequently provides the input to penetration testing.

4.4.41 The evaluation activities therefore not only address the relevant developer deliverables but also the evaluation products. For instance the Examine Construction Vulnerabilities activity requires that the evaluators examine the output from the Examine Strength Of Mechanisms activity as well as the developer's own List Of Known Vulnerabilities In Construction.

4.4.42 Sequencing dependencies are identified through the application of the following rules:

a) All activities which output or contribute to an evaluation product must be completed before any activity using that evaluation product can itself complete.

b) For any activity to be complete, it must have considered all relevant evaluation products (note that this does not rule out partially performing activities for later completion; for instance a subset of SEFs and components may be penetration tested before other SEFs have been checked at implementation).

4.4.43 For example, the activity Perform Penetration Testing cannot be completed until the activity Check Suitability Analysis activity has been completed (see figure 4.4.3). This is because the Check Suitability Analysis activity may identify construction vulnerabilities which the evaluators have to take into account during the Examine Construction Vulnerabilities activity. The evaluators' list of construction vulnerabilities will then be used to identify penetration tests that are carried out as part of the Perform Penetration Testing activity in order to ascertain whether the vulnerabilities are exploitable or not.

4.4.44 Using the dependencies identified in figure 4.4.2 and rules (a) and (b) above, a diagram can be constructed (see figure 4.4.4) showing the typical sequence for completion of the activities.

4.4.45 Figure 4.4.4 therefore represents a `generic EWP'.

Figure 4.4.2 Dependencies Between Activities

Activity / Intermediate Result	Threats	External Security Measures	SEFs	Security Relevant Events	Components (inc. basic components)	Security Mechs	Security Admin Fns	Critical Security Mechs	Constr. Vulns	SRFs	Oper. Vulns	Errors	Penetration Tests	Exploitable Vulns	ETR
Requirements	I O	I O	I O	I O		O					O	O			
Architecture		I	I	O	I O					O		O			
Design			I	O	I O	O		O		O		O			
Implementation			I	O	I O	I						O			
Development Environment	I *				I							O			
Operational Documentation			I	I	I		I O					O			
Operational Environment			I		I							O			
Suitability	I	I	I			I			O						
Binding			I			I			O						
SoMs						I		I O	O				O		
Construction Vulnerabilities	I	I	I			I		I	I	I			O		
Ease Of Use	I	I	I	I			I				O		O		
Operational Vulnerabilities	I	I	I			I		I			I		O		
Pen. Tests													I	O	
Write ETR	I	I	I		I	I		I	I	I	I	I	I	I	O

Key: * indicates that threats to the development environment may or may not be documented in the security target SEF: Security Enforcing Function
I indicates that an activity requires an intermediate result as an input..... SRF: Security Relevant Function
O indicates that an activity produces an intermediate result as an output

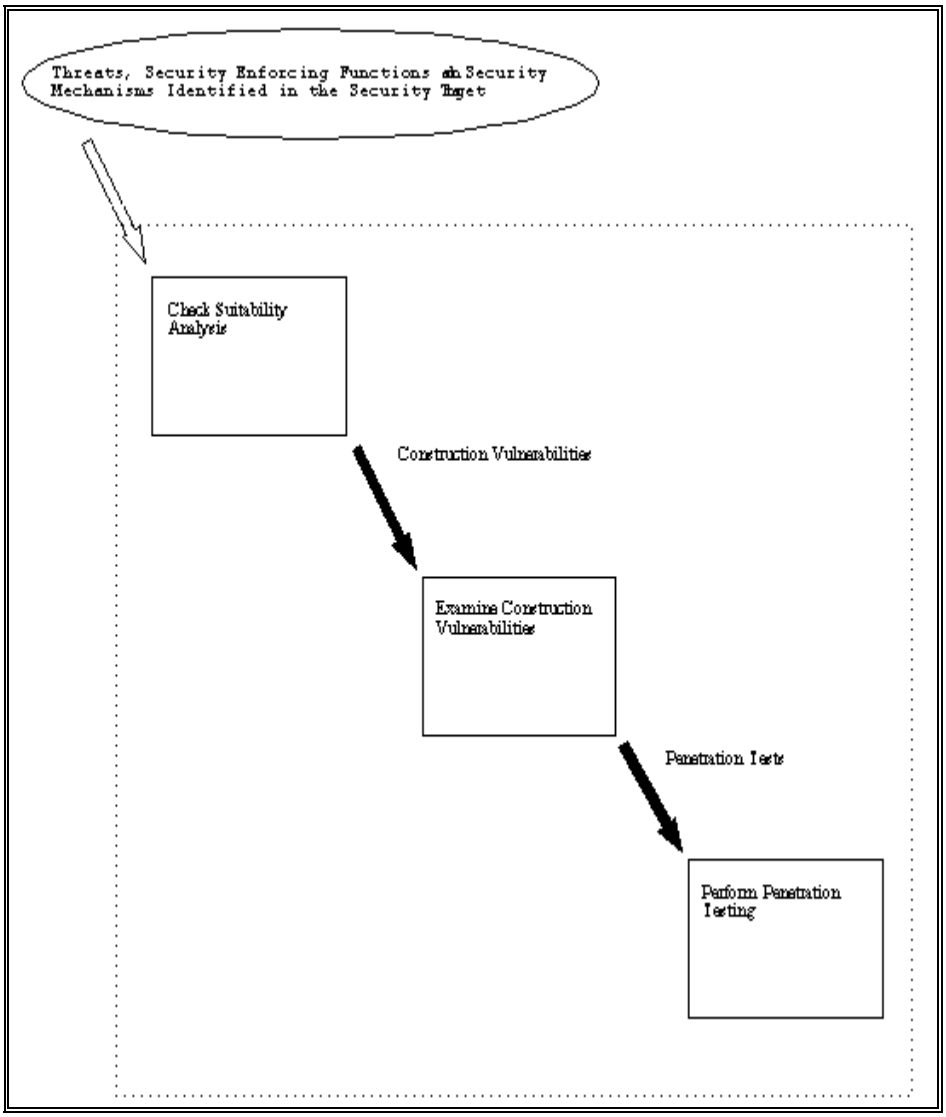


Figure 4.4.3 Example Activity Dependencies

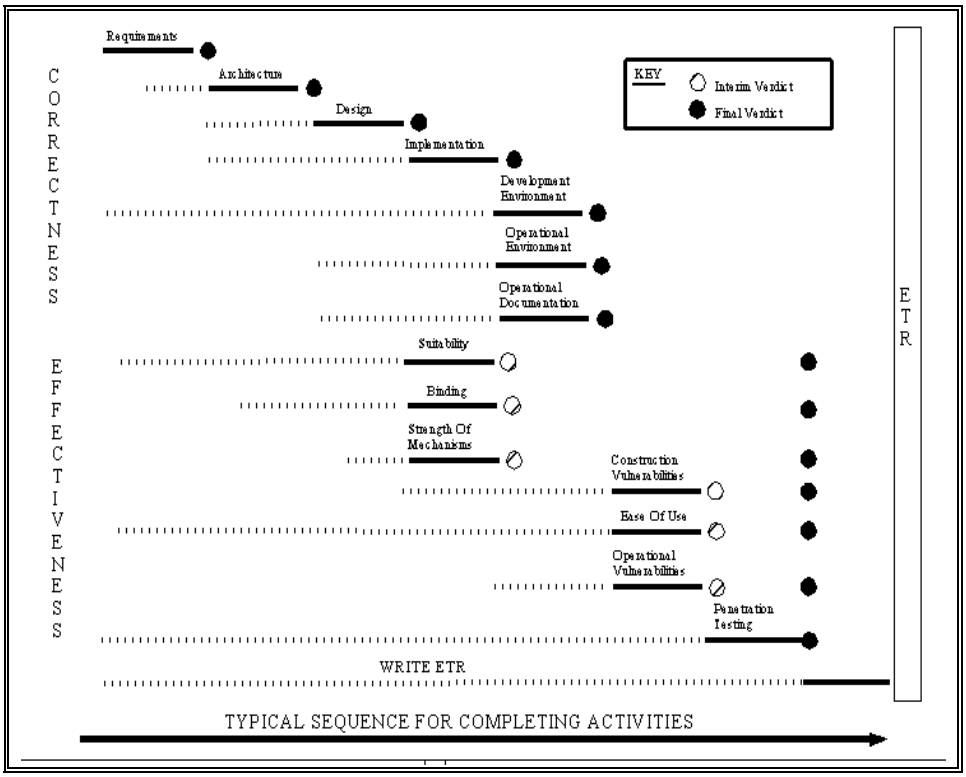


Figure 4.4.4 A Generic Evaluation Work Programme

Constructing Evaluation Work Programmes

4.4.46 Before starting on the conduct of an evaluation, the evaluators should be clear about what work is required. This implies a need for a plan of the work to be performed -i.e. an EWP.

4.4.47 In order to plan and report the evaluation work in a manageable way, the concept of a work package is required. The work package represents a unit of work that is carried out by the evaluators.

4.4.48 An activity may be divided into several work packages or several activities may be performed in one work package.

4.4.49 In developing specific EWPs from the generic EWPs discussed above, specific features of an evaluation will need to be incorporated into its plan. Such features include:

a) the target evaluation level and minimum strength of mechanisms rating;

b) specific names for deliverables; e.g. reference to the exact part of a document, or documents, which contain the architecture (see part 6, annex 6.A);

c) a specific set of abstraction levels for detailed design;

d) whether this is a re-evaluation and whether certified components of the TOE exist (this point is addressed in chapter 4.6);

e) whether the TOE is a system or a product;

f) the requirements for reporting, e.g. external meetings;

g) a deliverables list, giving a schedule of availability of deliverables from the sponsor;

h) whether re-evaluation information is required in the ETR.

4.4.50 The effect of the target evaluation level is profound. It is the evaluation level which determines the deliverables required by the evaluators, the content of the deliverables, and what the evaluators do to evaluate those deliverables.

4.4.51 For example, the ITSEC does not call for any analysis of source code at or below E2, and therefore the sponsor is under no obligation to provide this code to the evaluators (this is because some activities relevant to correctness will not be performed at lower evaluation levels).

4.4.52 On a small product evaluation to a low target evaluation level, on which the total evaluation effort is, perhaps, less than half a man-year over an elapsed period of two months, it might well be appropriate to have a single work package called *Examine The Architecture For Correctness*, or even one which combines this activity with *Do Binding Analysis* into a single package called `Architectural Assessment'.

4.4.53 By way of contrast, on a large distributed system to E6, where total evaluator effort could well be counted in many man-years, it would not be unreasonable to have a single work package simply for the one evaluator action of *Check That The Formal Arguments Are Valid*, since this may involve considerable checking of formal methods work.

4.4.54 An example of decomposing an activity into work packages would be that on a large system evaluation the *Perform Penetration Testing* activity might be implemented by work packages such as Prepare For And Specify Penetration Tests and Execute And Follow Up Penetration Tests.

4.4.55 Prepare For And Specify Penetration Tests would involve the administrative aspects (such as site access and office space - see part 6, annex 6.A) and the technical aspects of documenting a penetration test schedule and tests that conform to the requirements of the national scheme.

4.4.56 The Execute And Follow Up Penetration Tests work package would involve physically performing the documented penetration tests and recording the results. It may also involve following up any areas of suspected vulnerabilities in components of the system or re-testing components fixed as a result of uncovering vulnerabilities during previous penetration tests.

4.4.57 The issue of whether the TOE is a system or a product has technical repercussions (for instance checking the security target will involve slightly different work) and planning repercussions (for instance, a system may have several operational sites that need to be visited and penetration testing can only be performed at these sites within pre-planned timescales).

4.4.58 A product may be delivered to the ITSEF so that the evaluators have easy access to it for penetration testing; however, the operating environment and configuration to be evaluated would need to be emulated by the evaluators. Defining a `configuration for evaluation' is therefore particularly important for a product evaluation. This definition should be agreed between the evaluators and the sponsors. It must be documented in the ETR (see chapter 4.7).

4.4.59 The results of penetration testing are recorded in the ETR as described in chapter 4.7. They may also be stored in a local evaluation database. The storage of evaluation results is at the discretion of national schemes.

4.4.60 When a sponsor expects a TOE to be modified but wishes to maintain the certificate/certification report, he may request the evaluators to include information for **impact analysis** and re-evaluation in the optional chapter 7 of the ETR. This should be taken into account in the EWP.

Application of ITSEC

Introduction

4.4.61 The purpose of evaluation is to derive verdicts for a TOE's conformance to the ITSEC criteria. The certification process also produces verdicts: about whether the evaluation has conformed to the ITSEC/ITSEM, and whether the TOE has met the target ITSEC evaluation level.

4.4.62 This section is concerned with giving guidance to evaluators on applying the ITSEC to arrive at evaluation verdicts.

Evaluator Verdicts

4.4.63 A verdict results every time the evaluators complete an ITSEC evaluator action. A verdict can be one of *pass*, *fail* or *inconclusive*. For instance the action *check that the information provided meets all requirements for content, presentation and evidence* can produce verdicts as follows:

a) Pass if the information provided was found to meet all the requirements for content, presentation and evidence.

b) Fail if a noncompliance was found whereby the information provided did not meet all the criteria for content, presentation and evidence.

c) Inconclusive if the evaluators were unable to allocate a pass or fail verdict. This verdict indicates that the evaluator action is incomplete because, for example, the action was performed on a draft deliverable, or the evaluators could not understand part of the deliverable.

4.4.64 **If a fail verdict is arrived at, the evaluators must inform the appropriate bodies using the problem reporting mechanism**. It may be possible to agree with the sponsor some response which will address the problem. This, if successful, may cause the verdict to change.

4.4.65 All inconclusive verdicts must eventually be turned into pass or fail verdicts. If an inconclusive verdict is arrived at, the evaluators must agree with the sponsor some procedure for arriving at a conclusive verdict. This could involve waiting for a later representation; waiting for a later version of the same representation; or holding a meeting with the developer to discuss the technical issues involved. If no solution is found, then a fail verdict must be given.

4.4.66 A correctness verdict must be assigned to a development representation, e.g. the requirements or architectural design. A representation is assigned a pass verdict if all the evaluator actions performed on it produced a pass verdict. A representation is assigned a fail verdict if any evaluator action performed on it produced a fail verdict. An inconclusive verdict is assigned if the evaluator actions produced no fail verdicts, but did produce one or more inconclusive verdicts.

4.4.67 An effectiveness verdict must be assigned to every effectiveness aspect (e.g. ease of use or suitability). An interim verdict is derived from the evaluator actions performed upon the aspect in the same way as for correctness.

4.4.68 Note that each effectiveness aspect includes an evaluator action *check that the analysis has considered all of the information given in figure 4 for the evaluation level in question.* ITSEC figure 4 indicates that the operation representation must be considered for all evaluation levels. Also, penetration testing can reveal vulnerabilities against any of the effectiveness criteria. **Therefore, no final effectiveness verdict can be assigned until the penetration testing has been completed**.

4.4.69 The construction and operational vulnerability assessments require the evaluators to perform independent analyses, taking into account vulnerabilities found during evaluation. **This implies that these assessments cannot be completed until after the correctness evaluation**.

4.4.70 The interdependencies between evaluator actions are discussed in more detail in the *Generic Evaluation Work Programme* Section above.

4.4.71 The final conclusion of an evaluation is the result for the TOE as a whole. All inconclusive verdicts should be resolved (and replaced by pass or fail) for this to be arrived at.

4.4.72 It may be that a TOE can fail against the correctness criteria, but still be adequate for a lower evaluation level if that level does not include the criteria against which the TOE failed. In this case, it is possible to recommend the award of a lower level. If a TOE fails to achieve the claimed strength of mechanisms, it may be possible to award a lower strength of mechanisms. **If a TOE fails against any other effectiveness aspect, then the evaluators must fail the TOE**.

4.4.73 The developer can agree to amend a failed TOE or evaluation deliverables, including the security target. If the amendments are adequate, a fail verdict can be reversed.

Chapter 4.5 Evaluation Techniques and Tools

Objectives for this Section

4.5.1 Technical equivalence of evaluation results is supported by the use of standard procedures, including the use of a suitable evaluation technology. This implies the use of suitable techniques and tools.

4.5.2 During the evaluation process the evaluators can use evaluation techniques and tools in order to realise the EWP with the best cost/performance ratio, and in the shortest possible time. Use of such techniques and tools helps in achieving **objectivity**, repeatability and **reproducibility**. Part 6, annex 6.E gives general advice to evaluation tool builders.

4.5.3 The objective of this section is to describe both basic techniques, such as document reviewing, and animation, and also the techniques and tools that can be used to support them.

4.5.4 The objectives of this section are:

a) to cover the basic evaluation techniques;

b) to discuss the issues involved in selecting and using tools for evaluation;

c) to identify and describe different categories of techniques and tools and to indicate the likely relevance of each category to the evaluation activities.

4.5.5 It is not in the scope of the ITSEM to recommend particular techniques and tools; national standards may apply and the range of applicable techniques and tools is continually evolving. However, this section is provided to support the goal of technical equivalence of evaluation results.

4.5.6 No specific examples of techniques and tools are mandated as it is necessary to avoid any appearance of endorsing particular products.

Basic Evaluation Techniques

General

4.5.7 This section covers the basic techniques of evaluation. These can be used for a wide variety of evaluation levels and TOEs.

Informal Examination

4.5.8 The most basic evaluation technique is that of informal examination of documents.

4.5.9 This technique can be used for all the ITSEC *check* and *search* evaluator actions. With informal or non-machine-readable representations, there is little alternative to this method. There are certain dangers associated with the method, and the following guidelines are recommended to counter them:

a) Informal examination is not suitable for one person working alone for long periods of time because the quality of the results will deteriorate. Where possible, two evaluators should perform the work together.

b) **Evaluators performing informal examination must produce sufficient documentary evidence (e.g. intermediate results) to allow an assessment of their work to be made**.

Compliancy Analysis

4.5.10 A slightly more methodical technique is compliancy analysis. This is used to check two representations for consistency. The following steps are involved:

a) For each component of the higher representation, check (with the aid of the developer's traceability evidence where possible) that it is implemented correctly in the lower representation.

b) For each component in the lower representation, check that its existence is justified by the higher representation.

4.5.11 Again, the work is likely to be largely manual, in which case the guidelines above apply. In some cases, a database or hypertext system may be useful to keep track of the correspondence between representations.

4.5.12 **Evaluators must be able to demonstrate that they have considered every relevant part of the TOE in their analysis.**

Traceability Analysis

4.5.13 Traceability analysis is one of the underlying principles of evaluation. Traceability analysis is used to describe the notion of correct refinement of security enforcing functions through the representations of the TOE until its ultimate embodiment in the form of executable code and electronic circuitry.

4.5.14 Assurance in the TOE is built up by the evaluators checking that the security enforcing functions of the security target are correctly refined throughout the Architectural Design, Detailed Design, Implementation and Operation of the TOE. Assurance therefore increases progressively as further representations of a TOE are checked for traceability (hence the number of representations, and their detail, progressively increases with evaluation level).

4.5.15 A basic technique of evaluation is therefore to trace each security enforcing function through the various representations of the TOE, up to and including, the TOE's operation.

4.5.16 This was indicated in the *generic evaluation work programme* section earlier by the many intermediate results concerned with traceability.

The Review Process

4.5.17 The evaluation process involves a considerable amount of informal analysis. It is important to be able to demonstrate that this analysis has been done objectively. One way to do this is for each evaluator's work to be checked by others. This minimises the subjective element in the results. *The review process* described below is one approach to doing this checking, although a formal meeting may not be required.

4.5.18 **All evaluation outputs (see chapter 4.7) must be reviewed.**

4.5.19 The review process should involve at least the following roles:

a) the evaluation leader, who is responsible for the technical conduct of the evaluation;

b) the author, the evaluator(s) who performed the analysis;

c) the moderator, who is responsible for independently assuring that the review is performed properly.

4.5.20 Other personnel can be involved, e.g. technical specialists, certification body representatives, other evaluators (particularly if the author and evaluation leader are the same person).

4.5.21 The review process goes through a number of phases:

a) When an evaluation output is complete, the evaluation leader arranges for a review meeting at a time convenient to all the attendees. The time of the meeting should be such as to allow all the attendees to thoroughly study the output.

b) Before the meeting, the attendees study the output and check it for errors.

c) During the meeting, the attendees discuss the output and any errors found. The attendees consider whether any changes are required to the output. In general, the meeting should concentrate on deciding where changes are required, rather than on exactly what the changes should be.

d) At the end of the meeting, the attendees decide whether the output is acceptable as it is, whether minor changes are required, or whether major changes are required which must be reviewed in a further review meeting.

e) The decision is recorded. If a unanimous agreement cannot be arrived at among the attendees, the evaluation leader's decision will apply. Dissenting opinions should, however, be recorded.

Transcription

4.5.22 Transcription, or modelling, is the translation of a representation into another notation. For instance, a Z schema may be animated in Prolog in order that the evaluators can understand all the subtleties of the original representation. Often the act of transcription is at least as beneficial for the evaluators in terms of understanding as the end product.

Failure Analysis

4.5.23 A TOE which has availability requirements in its security target will be subject to a number of threats concerned with failure of the TOE. These threats can be either external hostile attempts to reduce the availability, or internal accidental threats resulting from the failure of the TOE itself.

4.5.24 External threats to availability can be considered during effectiveness assessment in the same way as external confidentiality or integrity threats. These analyses are concerned with the security functionality provided to counter an attack from outside the TOE.

4.5.25 Internal causes of availability loss need to be assessed using a technique which analyses the failure modes of the TOE. Such a technique is Failure Mode and Effects Analysis (FMEA) which is used in the reliability field to consider the reliability of equipments and systems. The FMEA technique is described in detail in a US military standard [MS1629A].

4.5.26 FMEA provides a standardised technique for considering all the failure modes of the TOE and their effect on the TOE's availability, taking into account any compensating provisions made in the TOE to counter the effects of failure.

4.5.27 The analysis identifies, for each component considered, its:

a) function;

b) failure modes and causes;

c) failure effects, at each representation;

d) failure detection method;

e) compensatory provisions;

f) resultant severity.

4.5.28 This technique can be used by evaluators to assess whether the TOE has adequate provisions to counter the threats to availability posed by its own failures.

Performing Performing Evaluator Activities

General

4.5.29 Figure 4.4.1 specifies all the activities to be performed by evaluators. These are described below, and suitable techniques proposed. However, not all evaluator actions are covered in detail.

Check Suitability Analysis

4.5.30 The main technique for this activity is informal examination.

Check Binding Analysis

4.5.31 The main technique for this activity is informal examination.

4.5.32 It may be necessary to search for covert channels in the TOE even if covert channels are not mentioned in the security target. Most techniques for covert channel analysis are based on the Shared Resource Matrix Method [SRMM]. Evaluators may find source code analysis tools, and matrix manipulation tools, of value when applying this method to a TOE.

Examine Construction Vulnerabilities

4.5.33 The main technique for checking the developer's analysis is informal examination.

4.5.34 **Evaluators are also required to perform their own analysis, based on vulnerabilities found during evaluation. This implies that evaluators must maintain a register of the problems found during evaluation. This must include problem reports, as well as minor correctness errors reported in a less formal way.**

4.5.35 The evaluators' analysis should address the following generic methods, which might be used to exploit a vulnerability:

a) change the predefined sequence of invocation of components;

b) execute an additional component;

c) use interrupts or scheduling functions to disrupt sequencing;

d) read, write or modify internal data directly or indirectly;

e) execute data not intended to be executed or make it executable;

f) use a component in an unexpected context, or for an unexpected purpose;

g) generate unexpected input for a component;

h) activate error recovery;

i) use implementation detail introduced in lower representations;

j) disrupt concurrence;

k) use interference between components which are not visible at a higher level of abstraction;

l) invalidate assumptions and properties on which lower-level components rely;

m) use the delay between time of check and time of use.

4.5.36 The *operation* correctness evaluator actions can indicate construction vulnerabilities as well as operational ones. For instance, consideration of user commands described in user documentation might reveal that the objective of a countermeasure is defeated if the commands are given in an unexpected order. The vulnerability is a construction vulnerability, rather than an operational vulnerability, because it makes use of properties of the TOE which were introduced during its construction.

Examine Strength of Mechanisms

4.5.37 This activity is mainly performed by informal examination.

4.5.38 Cryptographic mechanisms are not rated by ITSEFs (see ITSEC Paragraph 3.23).

4.5.39 Part 6, annex 6.C discusses strength of mechanisms.

Examine Ease of Use

4.5.40 This activity is performed mainly by informal analysis.

4.5.41 The evaluators are required to repeat any configuration procedure to check that the TOE can be configured and used securely, using only the user and administration documentation for guidance. This will require access to the TOE.

4.5.42 For a product, this is unlikely to be a problem. For a system evaluation, this may be difficult because the TOE may already be configured and installed and it may not be possible for the evaluators to repeat the process. In this case, it is necessary to distinguish between:

a) Those parts of the installation and configuration which are only performed once. For these, it should be sufficient to check that the configuration and installation of the TOE were performed correctly. **If such parts of a TOE are re-installed or re-configured, the TOE must be re-evaluated**.

b) Those parts of the installation and configuration which may be altered later. **For these, the evaluators must repeat the installation and configuration procedures**. However, these parts do not need to be re-evaluated if they are re-installed or re-configured.

4.5.43 Evaluators are required to *perform other testing where necessary to confirm or disprove the ease of use analysis*. This can be performed as part of the penetration testing activity.

Examine Operational Vulnerabilities

4.5.44 This is done using similar techniques to those of *Examine Construction Vulnerabilities.*

Check the Requirements

4.5.45 The main technique for this activity is informal examination of the security target.

4.5.46 At higher evaluation levels, evaluators can use animation tools to investigate complex parts of security targets.

4.5.47 When assessing the adequacy of formal descriptions at levels E4 and above, evaluators should consider the following questions:

a) Is the formal description at an acceptable level of abstraction, e.g. is it acceptable to describe a hardware gate in terms of classical logic (where the output of the gate is either 0 or 1) rather than ternary logic (where the output of the gate may be 0, 1 or undefined)?

b) Is the formal description expressed in an appropriate notation, e.g. is it acceptable to use, say, Z rather than CSP to describe a TOE which is composed of a number of concurrent, interacting processes?

c) Is the omission of any parts of the security target from the formal description justified, e.g. on the grounds that it is beyond the current state of the art for formal methods?

4.5.48 Security targets are discussed in part 6, annex 6.B.

4.5.49 **If the sponsor requires re-evaluation information in chapter 7 of the ETR, the evaluators must gather the necessary information during this activity.**

Check the Architectural Design

4.5.50 The main techniques for this activity are informal examination or informal compliancy analysis. At higher evaluation levels (i.e. E4 and above), modelling may be appropriate.

4.5.51 At the E6 evaluation level, the developer is required to prove consistency between the formal architectural design and the formal model of the security policy. This can be evaluated by informal examination, or it may be possible to use automated proof checking tools.

4.5.52 **If the sponsor requires re-evaluation information in chapter 7 of the ETR, the evaluators must gather the necessary information during this activity.**

4.5.53 **Where the criteria call for descriptions to be in semiformal or formal notations, the evaluators must check that the notations which are used, and their manner of use, are appropriate. To do so they must compare the notation with the requirements in ITSEC Paragraphs 2.65 to 2.80 (Specification Style).**

4.5.54 For example, in checking that the notation is acceptable as a formal notation, evaluators can consider the following:

a) Is the notation to a recognised standard, or otherwise academically respectable (e.g. Z, CSP, VDM and HOL)?

b) Is the notation otherwise acceptable to the certification body?

c) Can the sponsor demonstrate that the notation does indeed possess a well-defined syntax and semantics?

4.5.55 **The evaluators must verify that the language used for expressing the architectural design is capable of expressing features relevant to security.**

4.5.56 For instance, both data models and data flow diagrams qualify as semiformal notations. However, a single notation, such as a data model or data flow diagrams, may not be capable of expressing every facet of a TOE's architecture. In this case, the sponsor could make use of several notations which, in combination, will provide a complete picture of the architecture.

Check the Detailed Design

4.5.57 The main techniques for this activity are informal examination and informal compliancy analysis.

4.5.58 **Where a semiformal description of the design is delivered to the evaluators then they must check that the description corresponds to the level of formalism required by the ITSEC.** Acceptable semiformal styles include graphical representation (e.g. data flow diagrams, process structure diagrams) or a clearly defined restricted use of natural language (see ITSEC Paragraphs 2.72 to 2.75).

4.5.59 For instance, if Program Description Language (PDL) is delivered to the evaluators then they could ensure that the developer has a defined set of PDL standards which clearly define the syntax and semantics of the PDL constructs.

4.5.60 Where a developer uses Computer Aided Software Engineering (CASE) tools then the evaluators should confirm that the underlying method imposed by the tool is acceptable and conforms to the developer's quality standards.

4.5.61 At levels E5 and above, the ITSEC requires that *[the TOE]... shall incorporate significant use of layering, abstraction and data hiding.* These techniques are well-known in the safety-critical domain [ISO65A], and are intended to aid understanding and maintenance of the design. Note that, since use of these techniques is to be significant rather than all-encompassing, a single case of failure to use these techniques need not result in a failure verdict. **Rather, the evaluators must consider whether they find the TOE design clear.** If they do not, it is likely that layering, abstraction and data hiding have not been used adequately.

4.5.62 **Interface specifications must be checked for completeness and validity by cross-checking to the rest of the specifications of security enforcing and security relevant functionality.**

4.5.63 **Common variables, identified at levels E5 and E6, must be considered as a form of interface. Evaluators must ensure that these are defined once only and that the use of each common variable by a functional unit is justified and consistent with the definition of the common variable**. A checklist must be produced mapping common variables identified in the detailed design to references by functional units.

4.5.64 **If the sponsor requires re-evaluation information in chapter 7 of the ETR, the evaluators must gather the necessary information during this activity.**

Check the Implementation

4.5.65 Although this activity is more likely than any other to benefit from the use of automated tools, informal examination and compliance analysis can still be used.

4.5.66 If the implementation includes source code, static source code analysis tools can be used to assess the code quality and search for particular kinds of vulnerability. If the implementation includes hardware drawings, the developer's CAD tools may be useful in analysing them.

4.5.67 The evaluators are required to *use the library of test programs to check by sampling the results of tests*. This is one of two evaluator actions which allow sampling. **The following principles must be followed in choosing a sample:**

a) **The sample must provide a sample 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).**

b) **The sponsor and developer must not be informed in advance of the sample.**

c) **The size of the sample taken must be acceptable to the certification body.** To make the evaluation cost predictable, it may be possible to agree a sample size before the evaluation starts.

4.5.68 **Where sampling is performed, a rationale must be provided and the sample used must be recorded.**

4.5.69 This action may require use of the development systems. This should be taken into account in the initial contract with the sponsor.

4.5.70 The evaluators are required (at E2 and above) to *check that tests cover all security enforcing functions identified in the security target*. **As a minimum, evaluators must check that, for each testable statement in the security target, at least one test has been defined to demonstrate the statement.** The developer's justification (required at E4 and above) of *why the extent of test coverage is sufficient* may provide useful information for this.

4.5.71 The evaluators are required (at E3 and above) to *check that the tests cover all security enforcing and security relevant functions identified in the detailed design and all security mechanisms identified in the source code or hardware drawings.* **For the detailed design, the evaluators must check that every interface to every security enforcing or security relevant basic component has been used in a test.** The evaluators can check this with the aid of the developer's *correspondence between tests and the security enforcing and security relevant functions defined in the detailed design.*

4.5.72 Adequate coverage of source code has been achieved if the sponsor can demonstrate the following:

a) at E3, every statement of security-enforcing source code has been tested;

b) at E4 and above, every statement and branch of all source code belonging to a security enforcing or security relevant basic component has been tested.

4.5.73 *Source Code Coverage*: this box explains the testing of statements and branches. In order to test every source code statement of a sequential program, the developer needs to execute every source code statement in the program at least once. In order to test every statement and branch, the developer needs to execute every source code statement in the program at least once, and to execute every flow control construct in all possible ways.

4.5.74 The meaning of these requirements is best illustrated by an example. Consider the following program section:

```
if a
then B;
else C;
endif;
if d
then E;
endif;
```

4.5.75 In order to test every statement of this program, the developer needs to execute statements *B*, *C* and *E*, perhaps by running the program with conditions *a* and *d* true (executes B and E), followed by running the program with *a* false and *d* true (executes C and E).

4.5.76 In order to test every statement and branch of this program, the developer needs to execute statemements *B*, *C* and *E* as before, but he also needs to cover the case where *E* is omitted. This could be achieved by running the program with conditions *a* and *d* both true (executes B and E), followed by running the program with *a* and *d* both false (executes C only).

4.5.77 Test coverage of hardware drawings is a special case and a matter for national schemes.

4.5.78 The evaluators can use the correspondence between tests and security enforcing and security relevant functions in the detailed design and the correspondence between tests and the security mechanisms as represented in the source code or hardware drawings as an aid to checking coverage.

4.5.79 Certain compilers offer facilities for monitoring source code lines executed during tests. These facilities can be used by developers to produce test correspondences. The evaluators can use them to check test coverage, provided that the evaluators have confidence in the accuracy of these facilities.

4.5.80 The evaluators are required to *check all retesting following correction of errors*. This is concerned with regression testing, i.e. with retesting after correcting errors found by evaluators or developers.

4.5.81 The underlying principle here is that the corrections made should be consistent with the detailed design. If the TOE is changed after an error is detected, the changed components should be retested. In addition, some part of the system testing should be repeated to demonstrate that no secondary errors have been introduced.

4.5.82 **The evaluators must review the developer's statements about regression testing in the test plan to ensure that adequate retesting is performed. The evaluators must ensure that the regression testing strategy is followed.**

4.5.83 The evaluators are required to *perform additional tests to search for errors*. **The evaluators must therefore perform at least one additional test of each security enforcing function; the test must differ from the sponsor-supplied testing. Where this is inappropriate, a rationale for any reduction in testing must be given.** At E1 and E2 testing will be at the level of the security target.

4.5.84 **Additionally, at E3 to E6, an additional test of each security enforcing and security relevant function must be performed; the test must differ from sponsor-supplied testing. Where this is inappropriate, a rationale for any reduction in testing must be given.** Testing performed will be at the level of the detailed design and source code.

4.5.85 The evaluators, in performing these additional tests, may specify the tests and then enlist the support of the sponsor to perform the tests. **In this case the evaluators must witness the performance of the tests.** Alternatively the evaluators may decide to perform these functional tests as part of penetration testing (see below).

4.5.86 **Evaluators must also verify that the actual test results conform to the expected results.**

4.5.87 **If the sponsor requires re-evaluation information in chapter 7 of the ETR, the evaluators must gather the necessary information during this activity.**

Check the Development Environment

4.5.88 The evaluators should informally examine the development environment documentation.

4.5.89 At E2 and above, the evaluators are required to *check that the documented procedures are being applied*. The recommended method for doing this is to perform one or more visits to the development site. The purpose of such visits is to:

a) gain improved understanding of the development process by seeing it at work;

b) verify that the documented practices are applied in practice.

4.5.90 The visit should be agreed with the developer. Before the visit, evaluators should prepare a checklist of the topics they wish to cover. This can be given to the developer to allow him to prepare for the visit.

4.5.91 During the visit, the evaluators should interview development staff and audit configuration management and security practices.

4.5.92 The evaluators are required (at E4 and above) to use the developer's tools to create selected parts of the TOE and compare with the submitted version of the TOE. This is one of two actions which allow sampling.

4.5.93 **The evaluators must use every build process.** If the build process is uniform (all components are built in the same way) then only one component needs to be rebuilt. **If every component is built differently, the evaluators must rebuild every component.** It is unlikely that the evaluators will be able to build hardware components. **In the case of hardware components, the evaluators must witness the manufacture of such components at the development site.**

4.5.94 The evaluators may need to use the development system to perform this ITSEC action. This should be addressed in the contract with the sponsor.

4.5.95 File comparison tools can be used to compare the rebuilt component with the original. It should be noted that if the build process produces a timestamp in the component, this will be inconsistent with the original.

4.5.96 **If re-evaluation information is required in the ETR, evaluators must identify those development tools which are security relevant.**

Check the Operation Documentation

4.5.97 This is done by informal examination and reviewing. The evaluators familiarise themselves with the operational documentation and satisfy themselves that accurate information is given, sufficient to allow the TOE to be used and configured securely.

Check the Operational Environment

4.5.98 This is done by informal examination. The evaluators should familiarise themselves with the delivery, configuration, startup and operation documentation and satisfy themselves that accurate information is given, sufficient to allow the TOE to be maintained and operated securely. At E2 and above, evaluators will need to obtain information from the certification body about the required *procedure... which guarantees the authenticity of the TOE.*

Perform Penetration Testing

4.5.99 The following process, based on [LINDE], can be followed to select penetration tests:

a) the evaluators list all the errors, inconsistencies and vulnerabilities found during the evaluation;

b) the evaluators identify from the list those items which could result in a security breach, and are likely to be demonstrable in practice in penetration testing;

c) the evaluators prioritise the selected items, so that those which are most likely to be testable are performed first and those which are least likely to be testable are performed last.

4.5.100 **In order that the tests be repeatable, evaluators must produce a test script, describing the procedure for performing each penetration test, and the results expected from the test.** National schemes may include their own requirements for testing which is done away from the ITSEF (for instance, at developer sites).

4.5.101 Evaluators should inform the sponsor of their requirements for penetration testing. These may include:

a) adequate access to the TOE;

b) technical support from the developer;

c) accommodation, which may need to include secure storage facilities;

d) use of magnetic media.

4.5.102 Penetration testing may affect or damage the TOE. Evaluators should discuss with sponsors measures such as backups to minimise such damage.

4.5.103 Although most penetration tests should be performed to defined scripts, ad hoc testing (that is, testing without a pre-prepared test script) is allowed. **Such testing must, however, be justified and recorded in sufficient detail to make the test repeatable.**

4.5.104 Penetration testing can be aided by the use of security configuration and audit tools. These tools examine a system configuration and search for common vulnerabilities such as world-readable files and missing passwords.

Selecting and Using Evaluation Tools

Introduction

4.5.105 **Where an automated tool is used in arriving at an evaluation verdict, the ETR must record sufficient information about the tool and how it was used to allow the verdict to be repeated.**

4.5.106 **Any use of tools in this way must be acceptable to the certification body.** To avoid nugatory work, ITSEFs are advised to gain the certification body's agreement before using tools.

4.5.103 Certification bodies may, if they wish, maintain a list of automated tools which can be used to perform particular ITSEC evaluator actions.

Evaluation Tools

4.5.108 This subsection briefly describes the types of tools which can potentially be of use to evaluators.

4.5.109 *Animation tools*: these are used early on in a development to check high level representations such as the security target. Such tools can be exploited as follows:

a) convert the representation to be animated into an executable formal specification;

b) execute the formal specification in order to test the properties of the representation.

4.5.110 Practical experience suggests that producing the formal specification is at least as valuable as executing it.

4.5.111 *CASE tools:* Where a CASE tool has been used to produce a detailed design, the evaluators can attempt to use the tool to perform an independent validation of the design using the validation facilities that the tool provides. This can be performed using the tool for reviewing the design and using its reporting facilities to identify errors and omissions such as:

a) data flows which are not consistent between diagrams at different level of the design hierarchy;

b) data stores which have data flows entering but not exiting (i.e. the data is generated but not used by a process);

c) objects (e.g. data elements, data flows or data stores) which are defined but not referenced by a process;

d) references to undefined objects.

4.5.112 Detailed Design checking tools may be divided into:

a) *`Rule conformance' tools* : these verify syntactically and semantically, that the source code corresponds to its specification. These are often extensions of the source code checking tools.

b) *Proof tools* : these are able to perform a symbolic proof of partial or total correctness:

at a syntactic level: completeness, coherence, conformity.

at a semantic level: partial or total validity.

c) *Tracking tools* : these are able to analyse and report in a textual and graphical form the paths in an application program, to generate a procedure call tree and provide cross-referencing information. This category includes syntax analysers, semantic checkers, static analysers etc.

d) *Reverse engineering tools* : these are able to recreate and establish links between functions and specifications.

e) *Covert channel analysis tools* : the presence of covert channels may be checked for by the use of information flow analysis. This check would show that it was impossible for information to flow between processes in ways not specified.

4.5.113 *Source code analysis tools* may be divided into:

a) *data use analysers*: these check a source code program for incorrect usage of data, such as data items being read before written to;

b) *control flow analysers*: these search for control flow errors such as loops with no exit, or unreachable code;

c) *information flow analysers*: these examine dependencies between data items to search for unwanted dependencies;

d) *compliance analysers*: these compare the functionality of the source code with a formal specification, and attempt to prove compliancy.

4.5.114 *Object Code Analysis Tools:* at E6, the sponsor is required to provide the evaluators with tools to detect inconsistencies between source and object code. These tools can be used to investigate suspected inconsistencies.

4.5.115 *Build Tools*: starting with E3, well-defined programming languages are mandatory. Example 1(b) in part 5 of the ITSEM provides an example of how to perform the relevant evaluator actions. At E5 the source code of any runtime libraries shall be provided. Therefore compiler tools to handle that information are useful.

4.5.116 At E4, evaluators use the developers' own tools to rebuild selected parts of the TOE and compare the result with the submitted version of the TOE. It is in this sense that compilers and other build tools can be useful for evaluation. To this extent it is necessary for the evaluators to be familiar with the uses (and possible abuses) of such tools.

4.5.117 Should a compiler become a trusted part of the system (for instance if malicious software developers are mentioned as a threat in the security target), then it will be subject to evaluation in the normal way. This addresses in particular the subject of transitive Trojan Horses.

4.5.118 *Test Tools*: tools exist for certain compilers which can record the lines of source code which have been executed in a test suite. This can be used in providing evidence of test coverage.

4.5.119 The evaluators may need to develop software in order to perform penetration tests. Also, the evaluators may wish to obtain access to the developer's own test tools (such as test beds and monitoring tools).

4.5.120 **Any use of test tools must be documented in the ETR. Test software used by the evaluators must be archived.**

4.5.121 *Hardware analysis tools*: the evaluation of hardware requires some tool support not appropriate to evaluation of software. The differences centre around the use of CAD tools and the non-applicability of code analysis. The CAD tools can be viewed as design support tools (much like CASE tools) and all that has been stated above for design tools holds true for CAD tools. Note that it is unlikely that it will be possible to provide sufficient evidence of correct implementation without the use of CAD tools during development, except for extremely simple devices. CAD tools can provide the following facilities:

a) device libraries;

b) schematic capture (drawing package);

c) net list generation;

d) simulation;

e) Printed Circuit Board (PCB) design;

f) testing.

4.5.122 *Configuration and Audit tools*: for a number of widely-used operating systems, security configuration and audit tools exist. Such tools are likely to be useful during penetration testing.

4.5.123 A security configuration tool examines how an operating system has been configured, looking for known generic vulnerabilities such as world-readable files and guessable passwords.

4.5.120 A security audit tool examines an **audit trail** and searches for evidence of security breaches.

Summary: Recommended Techniques and Tools

4.5.121 Figure 4.5.1 gives an analysis of useful techniques for evaluators. The table was produced by considering the individual evaluator activities.

4.5.122 In the same way, figure 4.5.2 gives an analysis of useful tools for evaluators. Additional tools can be used if this would improve the reliability or cost of the evaluation.

Figure 4.5.1 Techniques for Evaluation

ITSEC Eval. Level	Evaluator Activities (only additional activities shown)	TECHNIQUES (only additional techniques shown)
E1	Check The Architectural Design Check The Implementation	informal examination or compliancy analysis functional testing penetration testing
E2	Check The Detailed Design Check The Implementation	informal examination or compliancy analysis test coverage analysis penetration testing
E3	Check The Implementation	source code test coverage analysis penetration testing
E4	Check The Requirements Check The Architectural Design Check The Detailed Design	semiformal examination or compliancy analysis examine formal model of security policy semiformal examination or compliancy analysis semiformal examination or compliancy analysis
E5	Check The Detailed Design	semiformal examination or compliancy analysis examine design layering, abstraction and data hiding
E6	Check The Architectural Design	formal examination or compliancy analysis
All	Check The Development Environment	informal examination development environment visits
All	Check The Operational Documentation	informal examination
All	Check The Operational Environment	informal examination
All	Check Suitability Analysis	examination
All	Check Binding Analysis	examination, including search for covert channels (where appropriate)
All	Examine Strength of Mechanisms	examination
All	Examine Construction Vulnerabilities	examination vulnerability analysis FMEA (where appropriate)
All	Examine Ease of Use	examination
All	Examine Operational Vulnerabilities	examination vulnerability analysis

Figure 4.5.2 Tools for Evaluation

ITSEC Eval. Level	Evaluator Activities	TOOLS (only additional tools are shown)
E1	Check The Implementation	test programs and tools (optional)
E2		
E3	Check The Implementation	test coverage tools (optional)
E4	Check The Requirements Check The Architecture Check The Detailed Design Check The Development Environment	animation tools (optional) developer's CASE tools (optional) developer's CASE tools (optional) developer's build tools
E5	Check The Implementation	source code analysis tools (optional)
E6	Check The Architecture Check The Implementation	proof checking tools (optional) tools to detect inconsistencies between source code and executable code (e.g. disassembler and/or debugger)
E3-E6	Check Binding Analysis	source code analysis and matrix manipulation tools (optional)

Chapter 4.6 Re-use of Evaluation Results

Introduction

4.6.1 An evaluation is a complex, resource intensive and time consuming process. The amount of effort to be spent and the money to be paid can be considerable, depending on the evaluation level targeted and the complexity of the TOE. In order to restrict the amount of necessary work, it is possible to make use of the results of previous evaluations:

a) for the evaluation of a TOE which includes one or more previously evaluated TOEs;

b) for the re-evaluation of a certified TOE after modification of the TOE, its security target or its deliverables.

4.6.2 This chapter advises evaluators on re-using evaluation results.

4.6.3 Chapter 4.3 addresses re-evaluation and re-use deliverables.

4.6.4 Part 6, chapter 6.3 and annex 6.D provide guidance to the sponsor/developer regarding impact analysis after modification of a certified TOE.

Overview

4.6.5 Examples of interest for reuse of evaluation results are:

a) products or systems which are composed of more than one product where at least one component has been evaluated as a product before;

b) products or systems that have been evaluated before and have been subject to change which makes a re-evaluation necessary (e.g. in the case of a new product release);

c) composition of products which have been previously evaluated to different evaluation levels (assurance profiles);

d) installation of a system which is made up of a previously evaluated product;

e) increasing the evaluation level of a previously evaluated product;

f). modification of a TOE, its security target, or one of the deliverables (e.g. new release of a product).

4.6.6 Generally, the extent to which it is necessary and useful to re-use evaluation results, depends on:

a) the use of the evaluated TOE;

b) the functionality of the evaluated TOE;

c) the evaluation level achieved;

d) the security target of the new TOE into which the evaluated TOE will be incorporated.

4.6.7 The generic guidance given to the evaluators in this chapter will focus on products or systems which are composed of more than one product where at least one part has been previously evaluated as a product.

4.6.8 The re-use of previously evaluated TOEs in a context different to that specified in the security target for the original evaluation is still largely a research area. The topic discussed here is closely related to problems in the area of system accreditation.

Generic Guidance for the Evaluator

4.6.9 In any case where there is doubt on how to apply the ITSEC criteria and no guidance is provided by ITSEM, guidance shall be sought from the certification body. This should, for example, be the case for TOEs which are composed of components evaluated at different evaluation levels.

4.6.10 Generally it is not possible to predict the evaluation level of a composition given the evaluation levels of its components. The composition might achieve a lower evaluation level than the minimum evaluation level of the components or even a higher level than the maximum level of the components given the required deliverables. This is due to the dependencies described in Paragraph 4.6.6. The strength of a combination might also depend on the type of security objectives as confidentiality, integrity, and availability.

4.6.11 Only after evaluation, in particular an effectiveness analysis by both the sponsor/developer and the evaluator, can assurance in the composition be obtained.

4.6.12 Different approaches have been developed for this problem. One idea is to use *functional partitioning* [TNI]. Another approach is provided by *partial orders on TCB subsets* [TDI]. The principle used in virtual machine systems is strict separation by a message passing kernel. The composition model presented in part 6, annex 6.F may also guide evaluators in performing re-evaluation work.

4.6.13 In the following paragraphs, ground rules will be presented which apply to TOEs which consist of at least two components where at least one is already evaluated to the same evaluation level as the composed TOE. If more than one component has been evaluated before, it is assumed that all these components have been evaluated to the same evaluation level.

4.6.14 As for every other TOE there has to exist a security target for the composition. A mapping from the security targets of the components to the security target of the composition shall be possible. This has to be checked as part of the suitability analysis.

4.6.15 The evaluation of the composed TOE with respect to the effectiveness criteria has to be performed under all circumstances.

4.6.16 The suitability analysis has to establish whether the security features of the individual components make up the stated security features of the composed TOE.

4.6.17 The binding analysis for the composed TOE shall be done just like the binding analysis during the evaluation of a non-composed TOE.

4.6.18 The evaluators have to check that the interface provided by a component is only used and can only be used in the composition such that the security features of the composed TOE are not compromised.

4.6.19 The construction vulnerability analysis shall be based on the 'use' relationship of the individual components. Internal details of the 'used' component must not compromise assumptions made for the 'using' component. It has to be assessed whether a potential vulnerability of a component is exploitable in the context of the composition. The list of vulnerabilities identified in the evaluation of a single component might contain vulnerabilities which are not relevant when the component is used in composition.

4.6.20 The ease of use analysis for the composed TOE shall be done just like the ease of use analysis during the evaluation of a non-composed TOE.

4.6.21 For a component already evaluated and used in a composition the correctness criteria for the development process do not have to be evaluated again. Evaluators can presume that the correctness verdicts are still true. This does not apply to tests concerning the effectiveness in the new context.

4.6.22 Correctness evaluation of the composed TOE as a whole is still required, even if the TOE consists entirely of pre-evaluated components. The security target, architecture, development environment and testing for the TOE as a whole should therefore be evaluated. Complete correctness evaluation according to ITSEC and ITSEM is required for the components of the TOE not previously evaluated.

4.6.23 If the composition concerns the user and administrator documentation, then the ITSEC criteria on operational documentation shall be applied.

4.6.24 The ITSEC criteria for the Development Environment Aspect 1 Configuration Control and those for the Operational Environment shall be applied as in the evaluation of a non-composed TOE. Nothing is to be done for these aspects with regard to the previously evaluated component.

Chapter 4.7 Outputs from Evaluation

Introduction

Objectives

4.7.1 This chapter provides a detailed description of the required output of an evaluation, i.e. the ETR and problem reports.

Scope

4.7.2 Chapter 4.4 described the production of evaluation reports as part of the evaluation process. This chapter is mainly concerned with the ETR that is produced by the evaluators for the sponsor of the evaluation and the certification body.

4.7.3 National schemes will require additional evaluation reports such as reports on evaluation methods, problem reports or reports on individual units of work. These are national scheme issues and are only addressed in this chapter where these affect the contents of the ETR.

Summary

4.7.4 Throughout this chapter it has been assumed that the ETR will be a single document, resulting from the activity *Write Reports* described in chapter 4.4. National schemes are not precluded from disregarding this assumption and implementing different arrangements.

4.7.5 For instance, later in this chapter it is stated that part of the ETR describes the security target for the TOE. A national scheme may make arrangements to incorporate the security target in the ETR or reference out to the security target (i.e. publishing the ETR together with the security target).

4.7.6 Another example would be the incorporation of an EWP to replace the chapter describing the evaluation work. The EWP should then at least contain the items as summarised below.

4.7.7 An ETR has the following objectives:

a) to describe the work actually performed during the evaluation;

b) to present the results obtained and conclusions drawn from this work.

4.7.8 The target audience for an ETR is:

a) the certification body;

b) the sponsor of the evaluation;

c) evaluators performing a re-evaluation.

4.7.9 In the case where the sponsor and developer are not the same, national schemes should also make arrangements for release of all or some of an ETR to developers considering re-using a TOE as part of another TOE. Provisions shall be made for the release of the ETR to another country. How this is arranged is a matter for national schemes.

4.7.10 The certification body is responsible for accepting an ETR.

4.7.11 This chapter identifies the minimal requirements for the *content and structure of the ETR* (via chapter and section headings) and discusses the contents of each chapter and section in turn.

Content and Structure of the Evaluation Technical Report

Preliminary Material

4.7.12 National schemes will prescribe the rules for marking and handling ETRs and will describe the form of the preliminary material in an ETR. For example, for government systems the ETR may be classified, for commercial systems and products the ETR may have to carry privacy markings.

4.7.13 Examples of preliminary material are:

a) disclaimers;

b) scheme logos;

c) copyright clauses.

Main Document

4.7.14 Figure 4.7.1 suggests a structure for an ETR. It is expected that this structure will be refined as evaluation experience increases.

4.7.15 The content of each chapter/section identified is described below. National schemes may choose to implement different structures for ETRs. However, the actual technical content must encompass the material below.

4.7.16 It must be noted that an ETR must provide the justification for all verdicts assigned by the evaluators. References to inaccessible material must not be made.

ETR Chapter 1 - Introduction

Background

4.7.17 This section contains an introduction to the background of the evaluation. It must include:

a) the evaluation identifier assigned by the certification body;

b) the name and version of the TOE evaluated;

c) the identity of the developer (including sub-contractors as applicable);

d) the identity of the sponsor;

e) the overall timescales of the evaluation;

f) the identity of the ITSEF.

Objectives

4.7.18 **This section must state the objective of the ETR (as outlined above).**

4.7.19 The objectives, at a more detailed level, are to:

a) present the evidence in support of evaluation verdicts and the evaluation conclusions;

b) support the re-evaluation of the TOE, if required by the sponsor.

4.7.20 Item b) above is particularly important if evaluations are to be efficient. It requires more information to be provided in the ETR than would be the case if item a) above was considered in isolation. Evaluators should remember this issue throughout the evaluation and especially when writing ETRs.

Scope

4.7.21 **This section must state that the ETR covers the entire evaluation. If this is not the case then a rationale must be provided.**

Structure

4.7.22 **This section must introduce the structure of the ETR**. Deviations from the structure of the ETR suggested in this chapter will be organised by national schemes.

ETR Chapter 2 - Executive Summary

4.7.23 This chapter provides the basis for any information regarding the results of the evaluation that is released by the certification body.

4.7.24 Where national schemes produce lists of certified TOEs, this chapter will form the basis for the information placed in those lists.

4.7.25 **The executive summary must not, therefore, contain information that is likely to be commercially or nationally sensitive in any way (the sponsor and certification body will confirm this when the ETR is accepted).**

4.7.26 **This chapter must contain:**

a) the identity of the ITSEF;

b) the actual evaluation level achieved;

c) the identifier of the TOE together with version number/release number;

d) a summary of the main conclusions of the evaluation;

e) the identity of the sponsor;

f) a brief description of the TOE;

g) a brief description of the TOE's security features.

ETR Chapter 3 - Description of the TOE

Functionality of the TOE

4.7.27 This section must contain a summary of the operational role of the TOE together with the functions that it is designed to perform, including:

a) the type of data to be processed (with sensitivity levels as necessary);

b) the various user types (linked to the above).

Development History

4.7.28 **This section must outline (for concurrent and, where possible, consecutive evaluations) the development stages performed in the production of the TOE.**

4.7.29 **All development methodologies, techniques, tools and standards relevant to the TOE production, and not covered in the results chapter, must be briefly outlined**.

4.7.30 **The deliverables to the TOE evaluation must be highlighted (with the detail such as issue status, dates, reference numbers and authors being subordinated to ETR Annex A).**

TOE Architecture

4.7.31 **This section must summarise the top-level design of the TOE. It must demonstrate the degree of separation between security enforcing and other components. It must outline the distribution of the security enforcing functions of the TOE through the hardware, firmware, software (and potentially the manual procedures as well) across the architecture of the TOE.**

4.7.32 All version numbers of components are listed in ETR Annex C.

Hardware Description

4.7.33 **The hardware description must give appropriate detail about all components at the architectural level that are relevant to the evaluation.**

Firmware Description

4.7.34 **The firmware description must give appropriate detail about all components that are relevant to the evaluation.**

Software Description

4.7.35 **The software description must give appropriate detail about all parts of the TOE software that are relevant to the evaluation. The description must link the software to the hardware and firmware components.**

ETR Chapter 4 - Security Features of the TOE

4.7.36 It is emphasised that an understanding of the contents of the security target is essential to the understanding of the ETR. Likewise, access to the security target and ETR together is necessary for re-evaluation to be efficient. **This chapter must either refer out to the security target or restate the security target in full.**

4.7.37 The content of this chapter is summarised below. More information is contained in the ITSEC (Chapter 2 and Annex A).

a) system security policy/product rationale;

b) specification of the security enforcing functions;

c) specification of the security mechanisms;

d) claimed rating of the minimum strength of mechanisms;

e) target evaluation level.

ETR Chapter 5 - Evaluation

4.7.38 Chapter 4.4 addresses the process for evaluation and for producing the actual EWP to be used. Chapter 5 of the ETR details the evaluation work performed, noting in particular any problems encountered (technical or managerial). The chapter is designed to assist the process of analysis within the Certification Bodies so that the overall evaluation process can be refined both technically and managerially (and therefore made more efficient and less expensive).

Evaluation History

4.7.39 This section is similar in design to the development history section of Chapter 3 outlined above. **It must outline the evaluation process utilised, and the key milestones that were:**

a) allocated at the start of the TOEs evaluation (for instance for the production of the EWP, ETR etc.);

b) actually met during the course of the evaluation.

4.7.40 Key milestones may include:

a) any evaluation startup meetings;

b) delivery of the security target;

c) when the penetration tests are performed;

d) any visits to the development or operational TOE's site(s);

e) completion of technical work.

4.7.41 **All evaluation methods, techniques, tools and standards used must be outlined.**

Evaluation Procedure

4.7.42 **A summary of the EWP must be provided in this section. The summary must include:**

a) the *evaluator actions* covered by the work programme, justified in terms of ITSEC;

b) the *work packages* performed (referring to ITSEM chapter 4.5 to demonstrate the use of acceptable procedures and ETR Annex D for detail) - this must highlight any differences between the work that was proposed in the EWP and that which was performed in practice together with a rationale for why these discrepancies existed;

c) a summary of how the evaluation deliverables (listed in ETR Annex A) mapped to the ITSEC *construction phases* - this must include any differences between the construction phases that were initially assumed and those that were actually delivered or used.

Scope of the Evaluation

4.7.43 **This section must identify exactly the components of the TOE evaluated and any assumptions made about components not examined.**

Constraints and Assumptions

4.7.44 **This section must highlight any constraints on the evaluation and any assumptions made during the evaluation.**

ETR Chapter 6 - Summary of Results of the Evaluation

4.7.45 **This chapter must provide summaries of the evaluation results in terms of the evaluator actions identified by the ITSEC.** The chapter's structure therefore, in the main, emulates the effectiveness and correctness chapters of the ITSEC.

4.7.46 **Subsection names corresponding to each evaluator action for each phase or aspect must be used.**

4.7.47 **Each subsection must reference the relevant work package reports contained in ETR Annex D.**

4.7.48 The first six sections are merely listed below. The final four sections (Penetration Testing, Exploitable Vulnerabilities Found, Observations Regarding Non-exploitable Vulnerabilities and Errors Found) are given with explanatory comments in the subsequent paragraphs.

a) Effectiveness - Construction

- Aspect 1 - Suitability of Functionality
- Aspect 2 - Binding of Functionality
- Aspect 3 - Strength of Mechanisms
- Aspect 4 - Construction Vulnerability Assessment

b) Effectiveness - Operation

- Aspect 1 - Ease of Use
- Aspect 2 - Operational Vulnerability Assessment

c) Construction - The Development Process

- Phase 1 - Requirements
- Phase 2 - Architectural Design
- Phase 3 - Detailed Design
- Phase 4 - Implementation

d) Construction - The Development Environment

- Aspect 1 - Configuration Control
- Aspect 2 - Programming Languages and Compilers
- Aspect 3 - Developers Security

e) Operation - The Operational Documentation

- Aspect 1 - User Documentation
- Aspect 2 - Administration Documentation

f) Operation - The Operational Environment

- Aspect 1 - Delivery and Configuration
- Aspect 2 - Start-up and Operation

Penetration Testing

4.7.49 As in chapter 4.5 penetration testing results have been treated separately because penetration tests are usually most conveniently executed as part of one work package.

4.7.50 **Any configuration options used during penetration testing must be recorded.**

4.7.51 **The results of penetration tests must be referenced to:**

a) the original work package where they were formulated;

b) the evaluator action prescribed by the ITSEC.

Exploitable Vulnerabilities Found

4.7.52 This section must describe the exploitable vulnerabilities found during the evaluation, identifying:

a) the Security Enforcing Function in which the vulnerability was found;

b) a description of the vulnerability;

c) the evaluator action being undertaken when the vulnerability was found;

d) the work package being undertaken when the vulnerability was found;

e) who found the vulnerability (developer or evaluator);

f) the date the vulnerability was found;

g) whether the vulnerability was fixed (with date) or not;

h) the source of the vulnerability (if possible).

Observations Regarding Non-exploitable Vulnerabilities

4.7.53 This section must report on non-exploitable vulnerabilities found during the evaluation (highlighting those remaining in the operational TOE).

Errors Found

4.7.54 **This section must summarise the impact of errors found during the course of development as perceived by the evaluators. Any actual results or conclusions, based upon the errors found, regarding the TOE's ability to meet the target evaluation level must be fully justified.**

ETR Chapter 7 - Guidance for Re-evaluation and Impact Analysis

4.7.55 This chapter is optional. It can be omitted if the sponsor has stated that he does not require any re-evaluation or impact analysis information.

4.7.56 **If present, this chapter of the ETR must record (by identifying the construction phase or aspect, or operation aspect, together with reference to evaluation deliverables):**

a) the classification of all parts of the TOE, at each construction phase examined, into one of security enforcing, security relevant or security irrelevant (defined in part 3 of the ITSEM);

b) the identification of those development tools of the TOE which are security relevant (defined in part 3 of the ITSEM);

c) any way in which the constraints or assumptions of the evaluation would impact re-evaluation or re-use;

d) any lessons regarding evaluation techniques or tools that would be useful for a re-evaluation (national schemes may decide to produce a separate document to record these);

e) all archiving details necessary for the evaluation to be restarted (national schemes may decide to produce a separate document to record these);

f) any specific skills that the `re-evaluators' are recommended to have before re-evaluation occurs (national schemes may decide to produce a separate document to record these);

g) the evaluators understanding of any way that the TOE could be configured such that the TOE becomes insecure.

ETR Chapter 8 - Conclusions and Recommendations

4.7.57 The conclusions and recommendations of the evaluation must be described in this chapter. The main conclusion will relate to whether the TOE has satisfied the security target, and is free from exploitable vulnerabilities.

4.7.58 Recommendations will normally be made to the certification body. It should be stated that these recommendations concern the parts of the TOE within the scope of the evaluation, and that there may be other factors that the evaluators are unaware of that may also influence the contents of the TOE's certificate/certification report.

4.7.59 The recommendations may include suggestions to other organisations, such as the sponsor or developer, to be forwarded by the certification body. These recommendations may include a reminder that the results of the evaluation are valid only for a particular version of the TOE when configured in a specific way, and that the certification body should be informed of any changes to the TOE as outlined in part 6, annex 6.D.

ETR Annex A - List of Evaluation Deliverables

4.7.60 **This annex must identify, with version numbers and dates received, all evaluation deliverables (generally the most recent version of a deliverable will suffice unless results are obtained from earlier versions) or make a reference out to the deliverable list.**

4.7.61 **Deviations from the deliverables identified in part 6, annex 6.A must be highlighted and justified.**

ETR Annex B - List of Acronyms/Glossary of Terms

4.7.62 **This annex must explain any acronyms or abbreviations used within the ETR. It must also define any terms used that do not appear within the ITSEC or ITSEM glossaries.**

ETR Annex C - Evaluated Configuration

4.7.63 **The configurations of the TOE examined during the evaluation (especially configurations used during the penetration tests, ease of use assessment and work relating to the operational TOE) must be clearly identified.**

4.7.64 **Any assumptions made or configurations not considered must be highlighted.**

Hardware Description

4.7.65 **The hardware description must give configuration information about all components at the architectural level that are relevant to the evaluation (and therefore to the security enforcing functions).**

Firmware Description

4.7.66 **The firmware description must give configuration information about all components (as described above) that are relevant to the evaluation (and therefore to the security enforcing and possibly security relevant functions).**

Software Description

4.7.67 **The software description must give configuration information for parts of the TOE software that are relevant to the evaluation (and therefore to the security enforcing and security relevant functions).**

ETR Annex D - Work Package Reports

4.7.68 This annex does not have to be produced if all of the work package reports are contained within ETR Chapter 6.

4.7.69 **If present, this annex must comprise the record of all the work performed (including sampling of results of tests done, techniques and tools used) necessary to justify the assignment of verdicts whilst performing the evaluator actions.**

ETR Annex E - Problem Reports

4.7.70 National schemes will instigate problem reporting procedures. **All problem reports issued must be incorporated into this annex.** Problem reports can be released before the evaluation is finished. **Problem reports must at least contain the following items:**

a) the evaluation identifier assigned by the Certification body;

b) the name and version of the TOE evaluated;

c) activity during which the problem was found;

d) description of the problem.

Figure 4.7.1 Structure of ETR (1 of 2)

ETR Chapter 1 - Introduction

- Background
- Objectives
- Scope
- Structure

ETR Chapter 2 - Executive Summary

ETR Chapter 3 - Description of the TOE

- Functionality of the TOE
- Development History
- TOE Architecture
  - Hardware Description
  - Firmware Description
  - Software Description

ETR Chapter 4 - Security Features of the TOE

- System Security Policy/Product Rationale
- Specification of the Security Enforcing Functions
- Specification of the Security Mechanisms
- Claimed Rating of the Minimum Strength of Mechanisms
- Target Evaluation Level

ETR Chapter 5 - Evaluation

- Evaluation History
- Evaluation Procedure
- Scope of the Evaluation
- Constraints and Assumptions

Figure 4.7.1 Structure of ETR (2 of 2)

ETR Chapter 6 - Summary of Results of the Evaluation

- Effectiveness-Construction
  - Aspect 1 - Suitability of Functionality
  - Aspect 2 - Binding of Functionality
  - Aspect 3 - Strength of Mechanisms
  - Aspect 4 - Construction Vulnerability Assessment
- Effectiveness-Operation
  - Aspect 1 - Ease of Use
  - Aspect 2 - Operational Vulnerability Assessment
- Construction-The Development Process
  - Phase 1 - Requirements
  - Phase 2 - Architectural Design
  - Phase 3 - Detailed Design
  - Phase 4 - Implementation
- Construction-The Development Environment
  - Aspect 1 - Configuration Control
  - Aspect 2 - Programming Languages And Compilers
  - Aspect 3 - Developers Security
- Operation-The Operational Documentation
  - Aspect 1 - User Documentation
  - Aspect 2 - Administration Documentation
- Operation-The Operational Environment
  - Aspect 1 - Delivery and Configuration
  - Aspect 2 - Start-up and Operation
- Penetration Testing
- Exploitable Vulnerabilities Found
- Observations Regarding Non-exploitable Vulnerabilities
- Errors Found

ETR Chapter 7 - Guidance for Re-evaluation and Impact Analysis

ETR Chapter 8 - Conclusions and Recommendations

ETR Annex A - List of Evaluation Deliverables

ETR Annex B - List of Acronyms/Glossary of Terms

ETR Annex C - Evaluated Configuration

ETR Annex D - Work Package Reports

ETR Annex E - Problem Reports

Part 5 Example Applications of ITSEC

Contents

Figures

Chapter 5.1 Introduction

Objectives for this Part

5.1.1 The objective is to demonstrate, through examples, how the ITSEM approach coupled with the ITSEC criteria can be applied to the evaluation of systems and products.

5.1.2 None of this part is mandatory. This part is intended only to illustrate the application of the ITSEC and ITSEM, and not to expand on them.

5.1.3 The ultimate goal is to provide complete examples of: .....

a) concurrent evaluations;

b) consecutive evaluations;

c) software;

d) hardware;

e) products;

f) systems;

g) **re-evaluation**;

h) **re-use** of evaluation results.

5.1.4 Items (d), (g) and (h) above are not covered in this version of the ITSEM but will be addressed in future versions.

5.1.5 Examples 1 - 6 are founded on European pre-ITSEC evaluation experience. Their origins are real life evaluations, but they have been sanitised and recast in ITSEC terms.

5.1.6 Example 7 is theoretical. It is speculative in nature.

5.1.7 Example 8 covers developer's security.

Relationship of this Part to the ITSEC

5.1.8 The examples provide coverage of:

a) examination of the development environment (at E2 and E4);

b) examination of the requirements for correctness (at E4);

c) examination of the architecture for correctness (at E4);

d) examination of the design for correctness (at E2);

e) examination of the implementation for correctness (at E2);

f) examination of the operation for correctness (primarily at E2, but with examples at all levels covering the concept of *state*, *describe* and *explain* in the context of a User Guide);

g) effectiveness assessment (at E3);

h) *examination of developer's security* (at E2 and E4).

5.1.9 Figures 5.1.1, 5.1.2 and 5.1.3 structure the evaluator actions into tabular form. The criteria are divided between actions for correctness assessment (figures 5.1.1 and 5.1.2) and actions for effectiveness assessment (figure 5.1.3).

5.1.10 The entries in the fields of the tables are references to paragraphs in [ITSEC].

5.1.11 The following convention has been adopted across the rows of the tables. A plus sign (`+') indicates that there are additional evaluator actions or that additional **deliverables** are required beyond those stated at the previous evaluation level. In other words, if there is no plus sign at some field, then the paragraph referred to is identical to that referred to by the entry to the left of the field.

5.1.12 In the case of effectiveness, the criteria are not stated separately for each evaluation level in the ITSEC. Effectiveness assessments are, however, performed with increasing rigour as the evaluation level rises, essentially because the depth of understanding which evaluators gain also increases with evaluation level.

5.1.13 The main body of this part presents eight examples. The coverage of each example is superimposed in figures 5.1.1, 5.1.2 and 5.1.3.

5.1.14 The headings of the examples identify the evaluation activity together with the target evaluation level.

5.1.15 In this part the term ***problem report*** refers to the formal recording of an **error** by the evaluators.

	E1	E2	E3	E4	E5	E6
Requirements Actions	E1.4	E2.4	E3.4	(2) E4.4+	E5.4	E6.4
Architectural Design Actions	E1.7	E2.7+	E3.7	(3) E4.7	E5.7+	E6.7+
Detailed Design Actions		(4) E2.10+	E3.10	E4.10	E5.10	E6.10
Implementation Actions	E1.13	(5) E2.13+	E3.13+	E4.13	E5.13	E6.13+

+ indicates additional rigour of action

indicates action covered in examples

(n) indicates covered in example n

Figure 5.1.1 ITSEC Evaluator Actions for Correctness (i)

	E1	E2	E3	E4	E5	E6
Configuration Control Actions	E1.17	(1a) E2.17+	E3.17	E4.17+	E5.17+	E6.17
Programming Language And Compiler Actions			E3.20+	(1b) E4.20	E5.20	E6.20
Developers Security Actions		(8a) E2.23+	E3.23	(8b) E4.23	E5.23	E6.23
User Dokumentation Actions	E1.27	(6a) E2.27	E3.27	E4.27	E5.27	E6.27
Administration Documentation Actions	E1.30	(6b) E2.30	E3.30	E4.30	E5.30	E6.30
Delivery And Configuration Actions	E1.34	(6c) E2.34+	E3.34	E4.34	E5.34	E6.34
Start-up And Operation Actions	E1.37	(6d) E2.37	E3.37	E4.37	E5.37	E6.37

+ indicates additional rigour of action

▨ indicates action covered in examples

(n) indicates covered in example n

Figure 5.1.2 ITSEC Evaluator Actions for Correctness (ii)

	E1	E2	E3	E4	E5	E6
Suitability of Functionality Actions	3.16	3.16+	(7) 3.16+	3.16+	3.16+	3.16+
Binding of Functionality Actions	3.20	3.20+	(7) 3.20+	3.20+	3.20+	3.20+
Strength of Mechanims Actions	3.24	3.24+	(7) 3.24+	3.24+	3.24+	3.24+
Construction Vulnerability Assessment Actions	3.28	3.28+	(7) 3.28+	3.28+	3.28+	3.28+
Ease of Use Actions	3.33	3.33+	(7) 3.33+	3.33+	3.33+	3.33+
Operational Vulnerability Assessment Actions	3.37	3.37+	(7) 3.37+	3.37+	3.37+	3.37+

+ indicates additional rigour of action

▨ indicates action covered in examples

(n) indicates covered in example n

Figure 5.1.3 ITSEC Evaluator Actions for Effectiveness

Chapter 5.2 Example 1, Examine the Development Environment (E2 and E4)

Introduction

5.2.1 This example presents two sub-examples (1(a) and 1(b)) which each address one development environment aspect at different evaluation levels.

Example 1(a) - Examine the Configuration Control Sub-activity (E2.17)

Introduction

5.2.2 This sub-example covers the development environment aspect 1, the configuration control actions. The characteristics of the evaluation were as follows:

a) the TOE was a real-time system;

b) the target evaluation level was E2;

c) the evaluation was performed concurrently with the system development.

Relevant Evaluation Deliverables

5.2.3 The inputs to this work were:

a) configuration list identifying the version of the TOE for evaluation;

b) information on the configuration control system.

Work Performed

5.2.4 The information on the configuration control system was contained in the developer's project configuration management procedures. These were examined by the evaluators (by reading and understanding them). In particular, the evaluators checked that:.....

a) the configuration list enumerated all basic components out of which the TOE was built;

b) the procedures required that all basic components and all relevant documentation shall possess a unique identifer and that identifer was obligatory in references;

c) the procedures required that the TOE under evaluation matched the deliverable documentation and that only authorised changes were possible.

5.2.5 The evaluators were subsequently able to visit the development site and confirm that the configuration control procedures were being applied by:

a) assessment of other delivered documentation for conformance to the practices;

b) interviewing staff to ascertain whether they are aware of the practices, and believed that they were being followed.

5.2.6 In order to ensure that the practices were applied consistently, the evaluators:

a) separately interviewed a number of development staff, asking the same questions at each interview;

b) interviewed staff in both senior and junior positions: while senior staff may be better informed as to what practices should be applied, junior staff may have a more realistic understanding of what is actually being done.

5.2.7 In order to further check that the documented procedures were being applied, the evaluators then:

a) selected several objects;

b) traced their change history through the configuration control system (checking areas such as proper approval for change, change request forms properly utilised etc.).

5.2.8 As the ITSEC correctness criteria for configuration control were met it was possible to assign a *pass* verdict.

Example 1(b) - Examine the Programming Languages and Compilers Sub-activity (E4.20)

Introduction

5.2.9 This sub-example covers the development environment aspect 2 - programming languages and compilers evaluator actions.

5.2.10 The TOE was implemented using a structured programming language and a commercially available compiler which possessed extensions to the ISO standard for that language. The target evaluation level was E4.

Relevant Evaluation Deliverables

5.2.11 The input to this work was:

a) the compiler reference manual;

b) the coding standards to be used by the development team, including a definition of the compiler options to be used.

Work Performed

5.2.12 The deliverables relating to the implementation language and compilers used in the development of the TOE were examined to determine whether the developer was using a well defined programming language. The evaluators noted that the compiler reference manual made no claims about compliance with a recognised standard for the language (e.g. ANSI or ISO standards).

5.2.13 The developer's own coding standards defined a subset of programming language statements which was derived from the ISO standard for that language. Because the compiler had not been validated against any recognised standard, the evaluators found it necessary to examine the compiler reference manual to check that the meaning of all statements identified in the developer's standards was unambiguously defined.

5.2.14 The compiler documentation was also examined to check the impact of the compiler options identified in the developer's standards. For instance, certain language compilers introduce unexpected effects (such as optimising source code statements out of loops) when the OPTIMISATION option is selected.

5.2.15 The evaluators noted that the compiler in question was a widely used commercial product, and as such, was itself well tested. Known compiler problems were well documented in the compiler release notes and were found not to have any consequences for the development of the TOE.

5.2.16 In addition to defining a subset of the language statements, the developer's coding standards excluded the use of constructs and techniques which the developer thought to be "unsafe". These included:

a) computed *GOTO*s;

b) aliasing (e.g. Fortran *EQUIVALENCE*).

5.2.17 The evaluators also noted that the developer's standards also enforced defensive programming practices, which included:

a) use of data types (enumerated types, subranges, etc.);

b) common definition of types and variables used by more than one component (e.g. by use of *INCLUDE* statements);

c) exception handling: range checking and array bounds checking, action on zero divide and arithmetic overflow;

d) type checking across separate compilation units.

5.2.18 The evaluators were able to confirm that the developer's coding standards were being adhered to. Checking adherence to the developer's standards was performed in parallel with the evaluator actions to examine source code as a part of the activity to examine the implementation for correctness.

5.2.19 Finally, the evaluators examined the "build files" and their use to ensure that the compiler options required by the coding standards were used across the development project.

5.2.20 In conclusion, the evaluators were able to assign a *pass* verdict to this aspect of the development environment

Chapter 5.3 Example 2, Examine the Requirements for Correctness (E4)

Introduction

5.3.1 This example covers the development process construction phase 1 - requirements evaluator actions. The TOE was a bespoke system. The security target for the TOE specified the F-B1 functionality class.

Relevant Evaluation Deliverables

5.3.2 The security target comprised:

a) System Security Policy (SSP);

b) System Electronic Information Security Policy (SEISP);

c) Security Policy Model (SPM);

d) target evaluation level of E4;

e) minimum strength of mechanisms rating of medium;

f) required cryptographic mechanisms.

5.3.3 The relationship between the above parts of the security target is described in figure 5.3.1.

5.3.4 The SSP, SEISP and SPM were consistent with ITSEC Version 1.2, Paragraphs 2.27 to 2.29 inclusive.

5.3.5 The SPM provided a formal model of the Identification, **Authentication** and Access Control requirements for the system in the Z notation. The SPM included pre-condition proofs to demonstrate that state transitions were secure. The SPM also provided an informal interpretation of the formally defined Identification, Authentication and Access Control requirements. The SPM referenced the Bell-La Padula model of the underlying security policy.

5.3.6 The SEISP security enforcing functions were considered to provide an informal interpretation of the security policy model in terms of the security target.

Work Performed

5.3.7 The requirements deliverables were checked for content and presentation and evidence. The evaluators found that, whilst the Identification, Authentication and Access Control requirements had been correctly specified in a semiformal style, the requirements for Accountability, Audit and Object Reuse had only been specified in an informal style.

5.3.8 The evaluators raised a problem report, recommending that those requirements that were not presented in a semiformal style should be described using dataflow diagrams, a logical data structure and entity life histories, consistent with the presentation style used for the security requirements correctly specified in a semiformal style (Structured Systems Analysis And Design Methodology (SSADM) had been adopted for the project).

5.3.9 The checking by the evaluators involved:

a) mapping the security enforcing functions in the SEISP to the security objectives and security threats identified in the SSP;

b) manual verification of the SEISP against the SSP to ensure the documentation was consistent;

c) validating the security enforcing functions of the SEISP against the formal SPM and the text within it;

d) verifying that the SPM preserved the intent of the Bell-La Padula model.

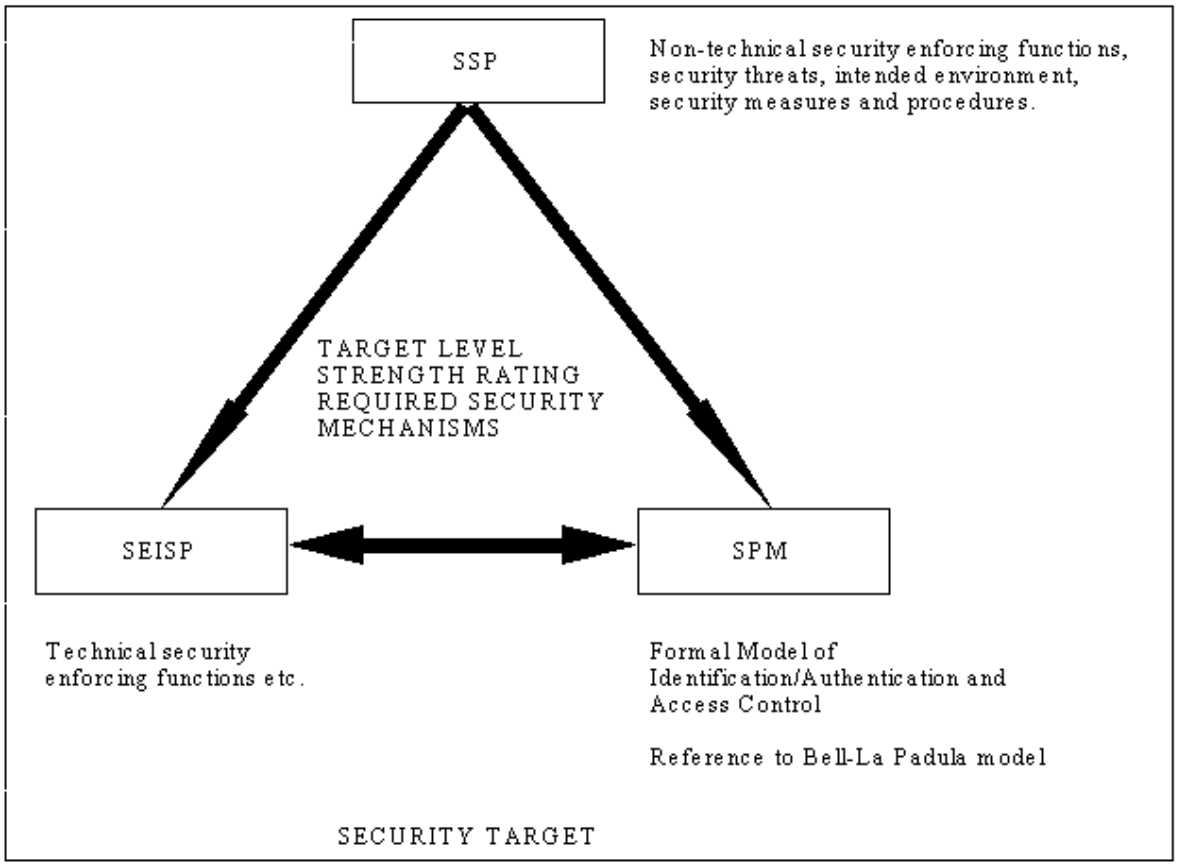
5.3.10 The SPM was validated by:

a) reading and thoroughly understanding the document;

b) understanding and independently verifying the proofs of predicate pre-conditions (to ensure that state transitions were indeed secure);

c) validating that the initial state was secure.

5.3.11 As the ITSEC correctness criteria for requirements were not clearly met it was possible to assign an inconclusive verdict. This inconclusive verdict was subsequently changed to a pass verdict when the evaluators were able to check the semiformal specification for Accountability, Audit and Object Reuse requirements supplied by the sponsor in a later stage of the evaluation.



Key:
SSP - System Security Policy
SEISP - System Electronic Information Security Policy
SPM - Formal Security Policy Model

Figure 5.3.1 Documentation Structure Breakdown

Chapter 5.4 Example 3, Examine the Architecture for Correctness (E4)

Introduction

5.4.1 This example covers the development process construction phase 2 - architectural design evaluator actions. The TOE was a distributed system comprising many components and had been developed just prior to the publication of the ITSEC. It was found that the documentation set provided fully met the requirements of E4 with some documentation which met the requirements of higher levels. The security target for the TOE specified the F-B3 functionality class with some additional functionality.

Relevant Evaluation Deliverables

5.4.2 The input to this work was the security target and the architectural design for the TOE.

5.4.3 Figure 5.4.1 provides a breakdown of the documentation delivered to the evaluators by the developers. The evaluators identified the parts of the documentation set which contained the architectural design. The architectural design is depicted on figure 5.4.1.

5.4.4 The architecture comprised:

a) System Functional Specification (SFS);

b) Formal Security Specification (FSS);

c) Security Architecture Document (SAD).

5.4.5 The TOE was developed using SSADM. The SFS comprised the outputs from SSADM Stages 1-3. The SSADM outputs were:

a) Dataflow Diagrams (DFDs);

b) DFD process descriptions;

c) descriptions of external entities (on the DFDs);

d) an I/O catalogue;

e) Logical Data Structure (LDS);

f) entity descriptions;

g) a data inventory;

h) an entity/data store cross reference;

i) an event catalogue;

j) an event entity matrix;

k) Entity Life History diagrams (ELHs).

5.4.6 The SFS provided the logical design, bringing together the functional requirements from the System Requirements Specification (SRS) and System Electronic Information Security Policy (SEISP). Part of the SFS, the Logical Man-Machine Interface (MMI(L)), presented the user view of the system (through state transition diagrams) for all the user types. The SFS made no commitment to physical aspects of the MMI such as screen layout.

5.4.7 Part of the SFS, the External Interface Specifications (EIS), defined the external interfaces (to external systems and existing embedded systems). The EIS outlined the external and embedded systems to be connected to the system network and gave details of communication interfaces. Security enforcing functions relevant to the communications interfaces were explicitly identified.

5.4.8 It should be noted that the MMI(L) and EIS were produced as separate documents. For the evaluation, however, they were considered to form part of the functional specification for the system.

5.4.9 The FSS comprised a formal specification, written in `Z', detailing a subset of the security enforcing functions, namely mandatory access control, accounting and audit. The specification included a textual expansion of the subset. A correspondence between the SFS and FSS existed. Although not a requirement at E4, the FSS was used to clarify which events specified in the SFS were relevant to security.

5.4.10 The SAD provided an overview of the intended TOE configuration and high level descriptions of how the security policy was to be implemented in the context of that configuration. It provided a description of how the security enforcing functions would be met. It described how separation requirements would be met.

5.4.11 The SAD specified the practices and procedures to be applied to the design and development life-cycle with respect to security enforcing, security relevant and non-security relevant parts of the TOE, such as:

a) quality procedures;

b) detailed design methods;

c) traceability documentation procedures;

d) functional testing;

e) configuration management;

f) change control.

5.4.12 These items were extracted and used as input to other evaluation activities (not part of this example):

a) Detailed Design (item (b), for background information);

b) Implementation (item (d), as it described the testing strategy to be undertaken and, in particular, stated the test coverage measures to be achieved and justified why the coverage would be sufficient);

c) Configuration Control (items (a), (c), (e) and (f) which explained the configuration management system in the overall context of the quality management procedures, identified and explained the use of configuration management tools, the acceptance procedure and the authority required to make changes).

5.4.13 One of the system components was a security enforcing and security relevant workstation. The SAD provided:

a) an overview of the workstation architecture;

b) identification of the security enforcing components (such as the interface to the network) and of the security relevant components.

5.4.14 Security enforcing functions were referenced in the SEISP, SFS (process descriptions) and FSS (textual expansion). The references were correlated in separate traceability documents, providing:

a) forwards traceability from SEISP security enforcing functions to SFS and FSS functions;

b) backwards traceability from SFS and FSS functionality to SEISP security enforcing functions.

5.4.15 The traceability documents provided justifications for non-traceable statements in the SFS and FSS.

Work Performed

5.4.16 All deliverables at the architectural design level were examined to check that the ITSEC requirements for content and presentation and evidence had been met. In particular, it was checked that:

a) All the intended external interfaces were identified and appropriate cryptographic and temporal separation mechanisms considered (in the EIS and SAD).

b) All the hardware and firmware components were identified and the functionality of the supporting protection mechanisms appropriate. For instance that the workstation would provide suitable object reuse arrangements for all memory (in the SAD).

c) The separation between security enforcing, security relevant and other components was realisable and sensible (in the SAD and SFS).

5.4.17 Project specific SSADM documentation practices were adopted. The project specific practices were examined to ensure that the evaluators had a clear understanding of the syntax and semantics of the semiformal notation. The SFS was compared to the documentation standards to ensure project practices were being implemented correctly.

5.4.18 The checking of the traceability evidence involved a manual verification of the SFS and FSS against the SEISP. The verification considered the introduction of non-traceable functionality, ensuring that the justification for this functionality was adequate.

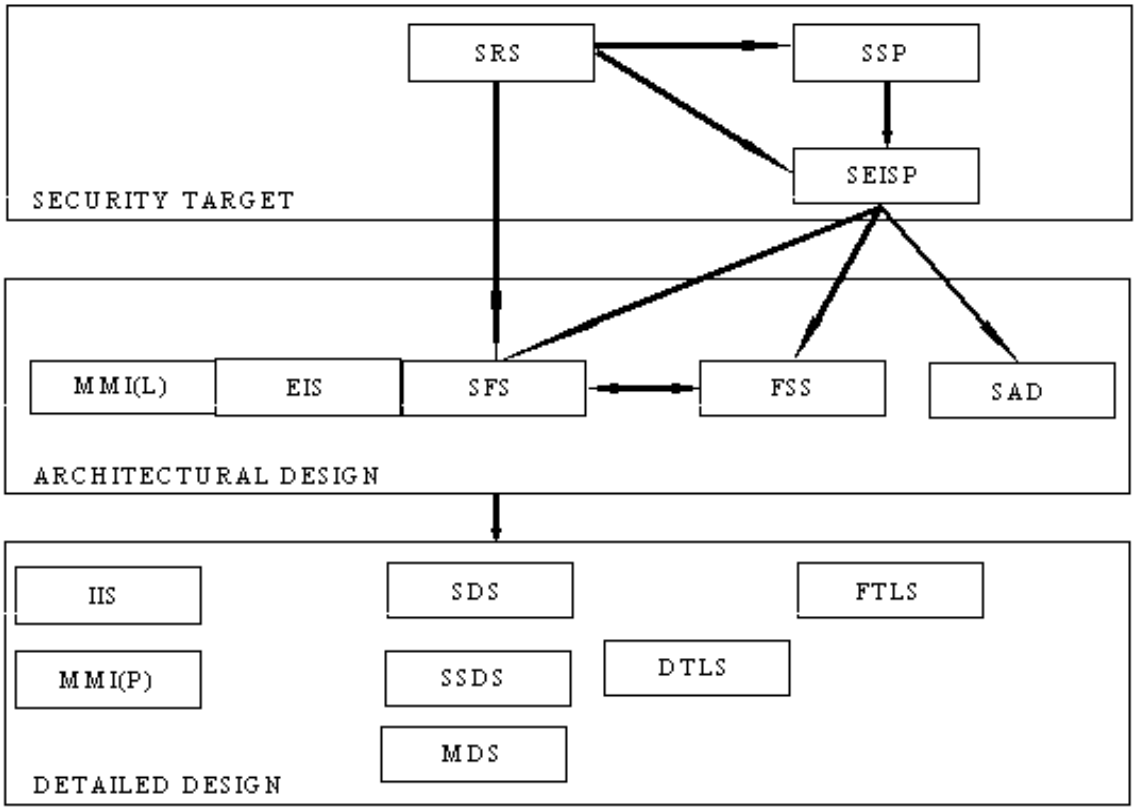
5.4.19 The evaluators found that the SSADM documentation in the SFS did not logically separate security enforcing, security relevant and other functionality and raised a problem report. The sponsor considered this issue and engaged a consultant, independent of the actual evaluation, to recommend a practical approach to resolving the problem.

5.4.20 The security consultants identified that the events described within the event catalogue had not been categorised as security relevant or non-security relevant. Through a consideration of both the FSS and the causes of the events the consultant was able to decide which events were security relevant.

5.4.21 The consultant recommended that a better approach which the developer might have adopted would have been to identify security functionality as one process on the top-level DFD. This could then have been refined into the security enforcing functions (such as Access Control, Accounting, etc.) and would have provided a clear logical separation of the functionality and also a clear indication of the independence of the security enforcing components. However, the consultant argued that at the E4 level the use of SSADM in the SFS was strictly correct and, if all the lower level DFD processes were categorised as being either security enforcing or non-security enforcing, the SAD document described enough physical separation for the architectural design to meet the requirements of the evaluation criteria.

5.4.22 This argument was accepted by the evaluators. The process categorisation was then supplied to the ITSEF enabling the evaluation to proceed.

5.4.23 As the ITSEC correctness criteria for the architecture were met it was possible to assign a pass verdict.



Key:
SRS - System Requirements Specificati on
SSP - System Security Policy
SEISP - System Electronic Information Security Policy
SFS - System Functional Specification (SSADM Stages 1-3)
IIS/EIS- Internal/External Interface Specifications

MMI(L) - Logical Man-Machine Interface
FSS - Formal Security Specification
SAD - Security Architecture Document
MMI(P) - Physical Man-Machine Interface
SDS, SSDS, MDS, FTLS, DTLS - Detailed Design

Figure 5.4.1 Documentation Structure Breakdown

Chapter 5.5 Example 4, Examine the Design for Correctness (E2)

Introduction

5.5.1 This example covers the development process construction phase 3 - detailed design evaluator actions. The TOE was a bespoke system. The security target for the TOE specified functionality class F-DI.

5.5.2 The evaluation was to be performed concurrently with system development and as a result, most deliverables were initially available only in draft form. Prior to evaluation, an ITSEF was contracted to review the draft documents to determine if they would satisfy the E2 requirements.

5.5.3 It became clear from the initial analysis that there were many areas in which the TOE ought to fail its target evaluation level. The sponsor was then informed of what steps should be taken to ensure a successful evaluation.

Relevant Evaluation Deliverables

5.5.4 The inputs to this work were the architectural design documentation (Level 1 design) and detailed design documentation for the TOE.

5.5.5 The detailed design documentation comprised:

a) subsystem design specifications (Level 2 design);

b) subsystem interface specifications (Level 2 design);

c) implementation specifications (Level 3 design).

Work Performed

5.5.6 The structure of the TOE levels of **representation** was assessed against the criteria for a clear and hierarchical relationship. It was found that subsystem interface specifications were produced as a Level 2 design document but contained information resulting from Level 3 design work. However, although presentation of the information did not match the evaluation level criteria, the information content was considered to be appropriate.

5.5.7 The mapping between security enforcing functions identified in the architectural design documentation and those stated in the subsystem design specifications was checked by means of manual cross-referencing to ensure:

a) all security enforcing functions were included in the detailed design;

b) the detailed design correctly preserved the intent of the architectural design.

5.5.8 A requirements traceability matrix was produced to check the mapping between security enforcing and security relevant functions stated in the subsystem design specifications and the security enforcing and security relevant components of the implementation specifications. Some errors were highlighted, in particular:

a) components in the implementation specifications identified by the evaluators as being security relevant were not stated in the subsystem design specifications (i.e. missing security relevant functionality in higher level representations);

b) security enforcing functions stated in the subsystem design specifications (e.g. object reuse) were not included in the implementation specifications.

5.5.9 The specifications of all security enforcing and security relevant mechanisms and components were checked to ensure that they were adequately documented. It was found that information needed to implement the security enforcing and security relevant components was not always present. For example:

a) insufficient level of detail on the use and content of key data structures and action to be taken on parameter validation failure;

b) missing external references (e.g. identification of libraries and external system components used);

c) natural language descriptions of security relevant functions inconsistent with the corresponding pseudo-code descriptions.

5.5.10 Interfaces to security enforcing and security relevant components were manually cross-checked against the system interface specification documents to ensure that all interfaces were identified and correctly specified. Errors were a particular concern because the system interface documents formed the programmer's definitive guide to the use of security enforcing and security relevant functions.

5.5.11 The above instances where the TOE detailed design representations were non-compliant with the evaluation level criteria were primarily a result of evaluating the TOE against draft representations. However, the evaluators also recognised that the detailed design was not being produced to be wholly consistent with the E2 requirements and problem reports were submitted by the evaluators.

5.5.12 The evaluators were unable to assign a pass verdict against the detailed design at this stage of the evaluation. An inconclusive verdict was assigned because the evaluators were examining draft representations of the TOE.

5.5.13 Final versions of the detailed design representations were re-examined before the completion of the evaluation and the above problems were found to have been resolved. A *pass* verdict was then assigned.

Chapter 5.6 Example 5, Examine the Implementation for Correctness (E2)

Introduction

5.6.1 This example covers the development process construction phase 4 - implementation evaluator actions.

5.6.2 The TOE was a bespoke system. The security target for the TOE specified functionality class F-DI.

Relevant Evaluation Deliverables

5.6.3 The inputs to this work were:

a) test documentation:

- package test specifications;
- Acceptance Test Plan;
- Activity Interface Test Specification;
- System Test Acceptance Specification;
- System Function Test Acceptance Specification;
- test schedules;
- test result files;
- description of test tools and user guide;

b) a library of test programs and tools used by the developers for testing the TOE.

5.6.4 Testing of the TOE was performed in two phases:

a) package tests;

b) acceptance tests.

5.6.5 Acceptance tests comprised the following stages:

a) activity interface tests: verifying that integrated components perform in the manner specified by the design and that integrity of shared data is maintained;

b) function tests: verifying that integrated components perform a system service as specified in the design and that user requirements have been met;

c) system tests: a full integration of hardware and software verifying that the system as a whole performs functions in accordance with the system design and meets the user requirements.

Work Performed

5.6.6 Assessment of the correctness for the TOE against the E2 criteria for implementation was carried out by performing the following for each phase and stage of testing identified above:

a) reviewing test documentation;

b) witnessing tests;

c) reviewing test reports;

d) repeating selected tests.

5.6.7 The developer's test strategy of the TOE demonstrated a controlled top-down approach, covering all security enforcing functions identified in the security target. The lowest level of testing to which the TOE components were subjected was package testing (a package being a collection of modules providing a related set of services) and not the individual modules. Nevertheless, this is sufficient for E2 as it is only necessary to show that the *tests cover all security enforcing functions ....*

5.6.8 The developer's objective for package testing was to test all functional threads identified in the design of the package with sufficient testing (confirmed by the use of in-line test harness code which could be selected at compilation time) to provide complete line coverage. Assessment of the integration testing, however, revealed that this objective was not met.

5.6.9 Acceptance test documentation was found to contain detailed descriptions of each test, including purpose, procedures and resources.

5.6.10 The acceptance test plan provided a high level requirements traceability matrix (RTM) which mapped test phases to user requirements. A more detailed RTM was provided in each of the acceptance test specifications. RTMs were manually checked to ensure that all security enforcing functions were adequately covered. It was noted that the RTMs were incomplete.

5.6.11 Acceptance tests for some of the security relevant functions were witnessed to ensure that test procedures were adhered to. This also involved requesting that previously completed tests were repeated.

5.6.12 Acceptance test reports were reviewed to ensure that each test had been completed successfully and to identify any weaknesses in the development of the TOE. The following commonly occurring problems were identified:

a) implementation modules which had (apparently) passed package testing failed to recompile in later test stages;

b) inadequate test coverage in package testing resulted in failures in later test stages;

c) system crashes occurred with undefined error status codes.

5.6.13 These problems highlighted weaknesses in the development process. The evaluators raised a single problem report to cover all of these issues and this problem report and the developer was able to subsequently demonstrate that these problems had been addressed.

5.6.14 Additional tests were identified by the evaluators to search for errors but due to the complexity of the test harness it was not possible for the evaluators to perform these tests themselves. To overcome this problem, the developer was provided with test specifications for additional tests to be carried out. These tests were then witnessed by the evaluators.

5.6.15 As the ITSEC correctness criteria for implementation were met it was possible to assign a *pass* verdict.

Chapter 5.7 Example 6, Examine the Operation for Correctness (E2)

Introduction

5.7.1 This example presents four sub-examples (6(a), 6(b), 6(c) and 6(d)) each of which addresses one operational documentation or environment aspect.

Example 6(a) - Examine the User Documentation Sub-Activity (E2.27)

Introduction

5.7.2 This sub-example covers the operational documentation aspect 1 - user documentation evaluator actions. The TOE was a system. The security target for the TOE specified the F-C2 functionality class.

5.7.3 The *Rigour And Depth Of Evidence* subsection at the end of this example interprets the transition in the rigour of the user documentation from *state* to *describe* to *explain*.

Relevant Evaluation Deliverables

5.7.4 The input to this work was the security target and the set of User Guides for the TOE.

Work Performed

5.7.5 The User Guides were mapped to the security enforcing functions in the security target to ensure a complete and consistent coverage of the security enforcing functions relevant to the end user. Administration functions such as audit were not considered relevant.

5.7.6 A number of site visits were undertaken by the evaluators. These enabled the evaluators to gain a good understanding of how the system operated. The operation of the system could then be compared to the descriptions in the User Guides (to assess their correctness) and residual doubts over the intent of the User Guides clarified.

5.7.7 The User Guides stated how the system menus could be used. The system menus were checked to ensure a correct correspondence to the User Guides.

5.7.8 The User Guides were thoroughly reviewed by the evaluators to ensure that possible security weaknesses were not identified to users.

Rigour and Depth of Evidence - Introduction

5.7.9 This subsection provides an example of how requirements on content, presentation and evidence change by level when the verbs state, describe and explain are used in the ITSEC, indicating good and bad points. The example chosen warns of the dangers of failing to take account of the context when interpreting this ITSEC concept.

5.7.10 The following changes in the requirements for content and presentation and evidence of the user documentation are detailed in the ITSEC:

a) At E1 and E2, *the user documentation shall **state** the security enforcing functions relevant to the end user and the user documentation shall **state** how an end user uses the TOE in a secure manner*.

b) At E3 and E4, *the user documentation shall **describe** the security enforcing functions relevant to the end user* and *the user documentation shall **describe** how an end user uses the TOE in a secure manner*.

c) At E5 and E6, *the user documentation shall **explain** the security enforcing functions relevant to the end user* and *the user documentation shall **explain** how an end user uses the TOE in a secure manner*.

5.7.11 The ITSEC Paragraph 0.12 defines state, describe and explain as the following. State means that relevant facts must be provided; describe means that relevant facts must be provided and their relevant characteristics enumerated; explain means that the facts must be provided, their relevant characteristics enumerated and justifications given.

5.7.12 The amount of effort in checking that the information provided meets all requirements for content, presentation and evidence therefore changes with evaluation level.

5.7.13 The security target for a system might specify a security enforcing function to limit logon attempts at a terminal. If so the requirements could be:

a) the system shall not allow more than three consecutive logon failures;

b) if three consecutive logon failures occur then the screen shall be blanked and the keyboard locked;

c) the system shall record all logon failures.

5.7.14 The effect of the different evaluation levels is discussed below.

Interpretation at E1 and E2

5.7.15 At E1 and E2, the user guides might state that a user only has three attempts to logon at a terminal and that after three failures the screen is blanked, the keyboard locked and each failure recorded by the system. This is a reasonable interpretation at E1 and E2.

5.7.16 The work performed at E1 and E2 would be as detailed in the *Work Performed* subsection above.

Interpretation at E3 and E4

5.7.17 At E3 and E4, the user guides might describe that a user only has three attempts to logon and that after three logon failures the logon process:

a) blanks the terminal screen by sending a control sequence (for example);

b) locks the keyboard by disabling its description record in the terminal configuration file and updates its internal terminal table (for example);

c) writes a message to the **audit trail** identifying the incident level, date, time, incident type (i.e. a logon failure), terminal identifier and user name entered. An example message is:

"WARNING: 12/08/91: 0935: LOGON FAILURE ON TTY03 BY J_SMITH" .

5.7.18 In this case the evaluators should point out that items (a) and (b) contain implementation details, which are irrelevant to the operational documentation. Item (c) contains detail that is not relevant to a User Guide, although it would make an adequate *description* for an Administration Guide.

5.7.19 Instead, the User Guide should describe the process of logging on and what happens. For example:

a) To logon to the system a user must first gain the attention of the operating system by pressing any key.

b) The system will then prompt for the user name which will be echoed.

c) The system will then prompt for the user's password. This will not be echoed.

d) If the user name and password is not a valid combination the system will display the message "ERROR: PLEASE TRY AGAIN".

e) The system will then re-prompt for the user name (step (b)). Three attempts are allowed. If the user is unsuccessful at the third attempt the screen will be blanked and the keyboard will lock. The terminal cannot be used for a period of five minutes (or as otherwise set by the system administrator).

f) If the logon is successful the system will then display the user's command menu.

Interpretation at E5 and E6

5.7.20 At E5 and E6, in addition to the above, the user guides might explain that:

a) blanking the screen gives the impression of a malfunction in order that the hacker gains no more information;

b) locking the keyboard prevents the hacker from trying more passwords;

c) auditing the event warns the administrator that a particular terminal (and also possibly a particular user account) is under attack.

5.7.21 Again, the evaluators should complain that this information is not relevant to a User Guide although it conveys the developer's justifications to a Security Administrator. It should be further noted that this example could also be criticised as providing useful information to a would-be attacker.

5.7.22 Paragraph 5.7.19 serves as a useful starting position to explain the logon process. In addition a paragraph along the following lines is required:

> *The purpose of logging on in this way is to assure the system that you are who you claim to be, in particular so that no-one else can logon to the system and pretend to be you. The purpose of allowing three logon attempts is to permit you to make an honest mistake in entering your password, but to stop an unauthorised user from attempting systematically to guess your password. The failure to logon within three attempts will be automatically brought to the attention of the system administrator.*

5.7.23 In addition to the work performed at E3 and E4, the justifications provided would also be checked against the security target explanations of security objectives, threats and security enforcing functions.

5.7.24 It should be noted that dependent on the target audience for the User Guide, the developers may provide more detail than required by the ITSEC. For example, the developers may explain things at E1 for inexperienced users; this may be a condition of a development contract.

Example 6(b) - Examine the Administration Documentation Sub-activity (E2.30)

Introduction

5.7.25 This sub-example covers the operational documentation aspect 2 - administration documentation actions. The TOE was a system. The security target for the TOE specified the F-C2 functionality class.

Relevant Evaluation Deliverables

5.7.26 The input to this work was the security target and the set of Administrator Guides for the TOE.

Work Performed

5.7.27 The Administrator Guides were mapped to the security enforcing functions in the security target to ensure a complete and consistent coverage of the security enforcing functions relevant to the system administrator.

5.7.28 During the site visits (see Paragraph 5.7.6) the evaluators learnt how the system could be administered by the system administrator. The operation of the system could then be compared to the descriptions in the Administrator Guides (to assess their correctness) and residual doubts over the intent of the Administrator Guides clarified.

5.7.29 For this particular system the ability of the administrator to access user information was severely restricted. Hence, the emphasis of the work was on ensuring that the procedures controlling the security parameters were sufficiently detailed.

5.7.30 Audit configuration was highlighted as an insufficiently described area. The procedures for installation of the system allowed the auditing mechanisms to be de-installed. The security target specified security enforcing functions for audit and hence the evaluators raised a problem report to ensure that the administration documentation was changed to state that the auditing mechanisms must be configured to run on the system.

5.7.31 The procedures detailing the identification and authentication mechanisms were scrutinised to ensure that:

a) the procedures for handling personal identification cards were consistent;

b) user accounts must be set up with a unique user name.

5.7.32 The procedures for the proper handling of backup and archive material were checked. A problem report was raised regarding the maintenance of all archive and backup material at the same site.

Example 6(c) - Examine the Delivery and Configuration Sub-activity (E2.34)

Introduction

5.7.33 This sub-example covers the operational environment aspect 1 - delivery and configuration actions. The TOE was a system. The security target for the TOE specified the F-B1 functionality class.

Relevant Evaluation Deliverables

5.7.34 The input to this work was the security target and the set of delivery and configuration practices for the TOE.

Work Performed

5.7.35 Delivery procedures were acceptable as they conformed to the guidelines published by the **national scheme**.

5.7.36 Each possible configuration identified in the procedures was checked to ensure that it did not compromise the security target.

5.7.37 A site visit was undertaken by the evaluators to witness the installation of the system. The system generation was witnessed to ensure conformance to the documented procedures and the audit trail checked to ensure that the trail recorded the actual system generation accurately.

5.7.38 As the ITSEC correctness criteria for delivery and configuration were met it was possible to assign a *pass* verdict.

Example 6(d) - Examine the Start-up and Operation Sub-activity (E2.37)

Introduction

5.7.39 This sub-example covers the operational environment aspect 2 - start-up and operation evaluator actions. The TOE was a system. The security target for the TOE specified the F-C2 functionality class.

Relevant Evaluation Deliverables

5.7.40 The input to this work was the security target and the set of secure start-up and operation practices for the TOE.

Work Performed

5.7.41 During the site visits (see Paragraph 5.7.6) the evaluators learnt how the system was started and operated. The operation of the system could then be compared to the descriptions in the practices (to assess their correctness) and residual doubts over the intent of the practices clarified.

5.7.42 No example results of self test procedures for security enforcing hardware components were available to the evaluators. An example of a security enforcing hardware component was a hardware filter linking a terminal identifier to the network. During the site visits self test procedures did capture a hardware problem with the equipment, which enabled the evaluators to gain some confidence in the self tests.

5.7.43 Security enforcing functions for accounting of start up existed. Therefore the sponsor provided examples of audit trail output created during start up and operation. This output was checked against actual start-ups to ensure a correct correspondence. Functional tests were scrutinised and provided a good coverage of the security enforcing functions.

5.7.44 The practices were thoroughly reviewed by the evaluators to ensure that possible security weaknesses were not introduced in the form of start-up options. The practices did not give special consideration to:

a) access to the computer room;

b) console output.

5.7.45 The procedures for handling of unmarked console output were not described and so a problem report was raised.

5.7.46 Procedures for removing and re-attaching a host from the network were not sufficiently detailed. During a site visit an operator, who followed the procedures as written, failed to remove a host correctly. This resulted in a violation of the site security procedures. The practices were therefore considered deficient in this area and a problem report was raised.

5.7.47 The ability of any user using the console to terminate security enforcing processes was insufficiently documented. A problem report was raised.

5.7.48 The ability to disable accounting mechanisms whilst the system was running was insufficiently documented in the practices. A problem report was raised.

5.7.49 As the ITSEC correctness criteria for start-up and operation were not met it was possible to assign only a *fail* verdict. The operational documentation was subsequently reworked by the sponsor and developer and then re-examined by the evaluators. It was then possible to assign a *pass* verdict against the ITSEC criteria for start-up and operation.

5.7.50 However, the evaluators considered that the disabling accounting mechanisms and the terminating security enforcing processes at the console may result in **potential vulnerabilities**. These problems were noted and examined as a part of the evaluators independent **vulnerability** analysis.

Chapter 5.8. Example 7, Effectiveness Assessment (E3)

Introduction

5.8.1 This chapter presents a complete worked example of the effectiveness criteria at E3. This example is entirely fictitious and is theoretic in nature. The example ignores the application of the correctness criteria. It should therefore be assumed that these have been applied at the appropriate time.

5.8.2 The primary objectives of this example are to illustrate:

a) how a sponsor can provide a well argued case for why known vulnerabilities are not exploitable in practice;

b) the work performed by the evaluators in independently checking the sponsor's vulnerability analysis.

5.8.3 Being a fictitious example, the vulnerabilities discussed in this example have been chosen merely to illustrate the analysis.

5.8.4 After the salient features of the security target and the architectural design of the example system have been described, the suitability and binding of functionality criteria are applied. These two criteria aim to identify vulnerabilities in the security target and architectural design respectively (see part 4, chapter 4.4). The security target used in the example is such that the application of the suitability of functionality criterion fails to identify any vulnerability. However, an architectural design has been chosen to illustrate a (partial) failure against the Binding of Functionality criteria.

5.8.5 Further example construction and **operational vulnerabilities** are then presented and the application of the Construction and Operational Vulnerability Assessments, Strength of Mechanisms analysis and Ease of Use criteria are illustrated. Due to the simplicity of the security target as presented in this example the Strength of Mechanisms and Ease of Use analyses are, however, limited.

Description of the Security Target

System Description

5.8.6 The example system is located in a large site belonging to a commercial organisation. The site is completely enclosed by a perimeter fence which is well guarded. All personnel are considered to be trustworthy. Visitors to the site are escorted at all times.

5.8.7 Within the site there are a number of areas which offer further protection in the form of physical access control and other procedural security mechanisms. There is a low TEMPEST and cryptographic threat. Terminals are located in secure rooms and personnel will be prevented, by the authorised users, from using a terminal unsupervised in a room that they are visiting.

5.8.8 The site contains a variety of different IT systems, procured at different times from different manufacturers and used for different purposes, including transaction processing, billing and company administration.

5.8.9 Each such system, referred to as an end system, can be identified by a system number, S#, and a security level, SL. They each provide the requisite level of security for their own needs (e.g. some information is Management-in-Confidence, MiC). Because of the physical site security, the numbers of users per end system, the perceived threats and the sensitivity of data, no end system warrants more protection than provided by an ITSEC class F-C2 operating system.

5.8.10 For the purposes of this example each end system, in general, will possess its own I&A and DAC functions under the direct control of the system manager for the end system in question. Moreover, all end systems can be regarded as star-networks, bearing a central host (or cluster) serving a closed community of users primarily using dumb terminals.

5.8.11 Connections between user terminals and host computers, which may be located in different buildings, were once made via dedicated fibre-optic cables. These are now replaced by a site-wide area network (SWAN), the subject of this example. The SWAN is a TCP/IP token ring network consisting of a dual counter rotating backbone and various subrings. End system equipment is attached to the SWAN by host access points (HAPs) or terminal access points (TAPS). TAPS are fed via RS232 connections to terminal servers (TSs) and thence by Ethernet connection to routers principally connected to the subrings, although some connect directly to the main backbone. The HAPs attach directly to a router. These are permanent connections. When a terminal is switched on, the SWAN automatically initiates a log-on sequence. If user log-on is successful the user is presented with a menu of permitted services at his/her terminal. This menu will list those hosts to which that user has been permitted access by the SWAN manager. A virtual circuit is then established between that terminal and the chosen host. The user then has to log-on at the host.

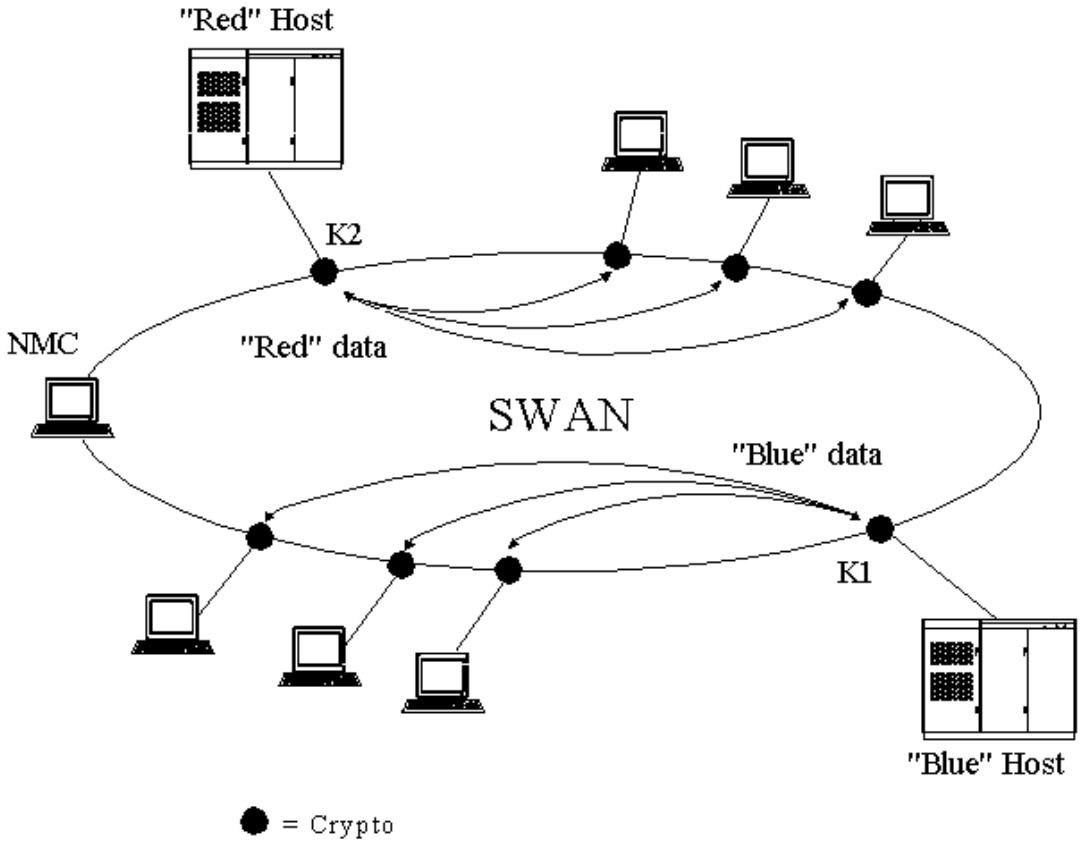
5.8.12 Security profiles (what hosts a user is permitted access to), and other security mechanisms are administered on the SWAN through one of two network management centres (NMCs), which provide I&A, DAC and MAC for the SWAN.

Security Objectives

5.8.13 The SWAN therefore provides a connectivity service to end system host computers and their user communities. It has two security objectives:

a) to protect against unauthorised access to an end system (S1);

b) to protect the confidentiality of information in transit (S2).



The SWAN provides connectivity between different computer systems and their users. *End system access control* is provided by the network management centre (NMC) in that it only grants access to authorised services on the network. A user, say of the blue system would have to logon to the SWAN, select an authorised service and then logon to the host. It is a requirement to keep different user communities separate, and since these might be operating at different levels (e.g. the red data might be Management-in-Confidence (MiC) and the blue data unclassified), then this is a mandatory requirement. *Communications confidentiality is* provided by end to end encryption, with unique keys per end system (i.e. K1 and K2 in the figure). Physical and procedural controls are used to ensure that users could only use their own terminals (in other words a user of the red system would not be allowed to enter the terminal room of the blue system and use one of their terminals).

Figure 5.8.1 Architectural Design of the SWAN

Threats to Security

5.8.14 The threats to the security of the SWAN are:

a) A user could masquerade as another user, when accessing the SWAN (T1).

b) A user could request, or otherwise attempt to use, a service for which he/she has no authority to use (T2).

c) A person could eavesdrop on, or otherwise capture data in transit across the network (T3).

d). A user could masquerade as another user, when accessing a host (T4).

Security Policy

5.8.15 The security policy calls for three forms of access control: MAC, HAP-sharing (see below) and DAC.

5.8.16 MAC is satisfied if, and only if:

$$S\#_H = S\#_T$$
$$SL_H = SL_T$$

where (S#H, SLH) and (S#T, SLT) are the system numbers and security levels of the host computer and terminal respectively.

5.8.17 The HAP-sharing policy takes account of the case where a terminal may be offered access to more than one host (H1...Hn), and requires:

$$S\#_{H1} = S\#_{H2} = ... \ S\#_{Hn} = S\#_T$$
$$SL_{H1} = SL_{H2} = ... \ SL_{Hn} = SL_T$$

5.8.18 Within these constraints, DAC permits only those items of end system equipment which that end system manager wishes to be connected.

Security Functionality

5.8.19 Security in the SWAN is enforced by four **countermeasures** (CM1..CM4):

a) an I&A function is used to authenticate users logging onto the network (CM1);

b) the access control policy described above is enforced by the NMCs (CM2);

c) approved encryption/decryption devices are placed between the end system equipment and the HAPs and TSs (CM3);

d) an I&A function is used to authenticate users logging on to a host (CM4).

5.8.20 The cryptographic devices located between a host computer and the SWAN always operate in cipher mode - there is no clear text bypass. Those placed between terminal equipment and the network, however, do have a bypass mode. Initially such a device works in bypass mode. This permits interaction with the network in clear (i.e. messages sent to and from the network are not encrypted). Once the user has established a connection to the host computer, the host computer crypto-unit (HCU) transmits a signal to the terminal crypto-unit (TCU) which switches the TCU from bypass to cipher mode. On termination of the session between the user and the host, the circuit is broken and the TCU reverts to bypass mode. A lamp on each crypto unit is illuminated while the unit is in cipher mode and is extinguished when it switches to clear mode.

5.8.21 Keys are managed externally, i.e. they are the responsibility of the end system managers and not of the SWAN manager. There is one crypto-key per end system and no two such keys are the same.

5.8.22 The discretionary access control policy is enforced by the NMCs.

5.8.23 No router or terminal server software is considered to be security relevant.

Required Minimum Strength of Mechanisms

5.8.24 The required minimum strength of mechanisms is *medium*. Consequently (see part 6, annex 6.C), the maximum level of opportunity, expertise and resources available to an attacker is considered as being *medium*.

5.8.25 The developers choose to use a cryptographic mechanism rated by the appropriate national authority of at least *medium*, an Access Control mechanism (for CM2) rated *high*, and *basic* I&A mechanisms (for CM1 and CM4).

5.8.26 The security target puts forward the argument that the cryptographic mechanism is the critical mechanism for the SWAN since if the access control mechanisms fail the attacker merely gains access to encrypted data thereby continuing to uphold both security objectives.

Configurable items

5.8.27 There is a basic crypto unit which can be configured to be a HCU or a TCU by the insertion of a keycard which contains an application program and the crypto key. The cards are of different size and colour, and one installed part of the keycard slot for HCUs is blanked off mechanically so that it will not physically accept TCU cards. This feature readily distinguishes HCUs and TCUs.

5.8.28 Passwords can be configured to be of between 8 and 12 characters in length and are auto generated. The life-time of a password can be configured to be between 1 and 60 days. Both ranges are specified in the security target.

5.8.29 No other security functions are configurable.

Effectiveness Analysis

Suitability Analysis

5.8.30 The ITSEC requires the sponsor to provide a **suitability analysis**, linking the security enforcing functions and mechanisms to the identified threats that they are designed to counter, and showing that those threats are adequately countered.

5.8.31 In this example, the sponsors's suitability analysis considers each threat listed in Paragraph in isolation of the other threats. This analysis identifies at least one function or mechanism that can counter the threat. The sponsor does not take into account the composition of mechanisms, i.e. the sponsor does not consider the architectural design of the SWAN (figure 5.8.1), only the bald enumeration of threats and countermeasures as given in the security target.

5.8.32 In this example, the sponsor demonstrates the direct correspondence between countermeasures and threats to security enforcing functions as follows:

a) masquerading as someone else when attempting to gain access to the SWAN (*T1*) and the SWAN logon function (*CM1*);

b) requesting, or otherwise gaining access to an unauthorised service (*T2*) and the SWAN access control function (*CM2*);

c) eavesdropping or otherwise capturing data in transit across the SWAN (*T3*) and the crypto function (*CM3*);

d) masquerading as someone else when attempting to gain access to a host (*T4*) and the host logon function (*CM4*).

5.8.33 The sponsor's suitability analysis includes the table shown in figure 5.8.2, demonstrating the correspondence between security objectives, countermeasures and threats.

Figure 5.8.2 Suitability Analysis

Security Objective	Countermeasure	Threat
S1 - Host Access Protection	CM1 - SWAN Logon	T1 - masquerading SWAN access
S1 - Host Access Protection	CM2 - Host Access Control	T2 - requesting or gaining access to non-authorised service
S1 - Host Access Protection	CM4 - Host Logon	T4 - masquerading to host
S2 - Network Confidential	CM3 - Crypto	T3 - capturing data

5.8.34 Both the SWAN logon and host logon functions are proprietary secret password systems. In the sponsor's suitability analysis, it is argued that both of these functions are suitable since an attacker would have to know the secret password of the other person. It is also argued that:

a). the SWAN access control function is suitable because it will only allow an identified user to choose between services for which that person is authorised;

b) the crypto function is suitable because the host crypto unit has no bypass mode and always transmits data enciphered with an appropriate algorithm and key unique to that host and known only to the authorised users of that host.

5.8.35 Consequently, only an *authorised* user who eavesdrops or otherwise captures data in transit across the SWAN should be able to decipher the data. In conclusion, therefore, all of the security enforcing functions are suitable.

5.8.36 Notice that these arguments are not concerned with the strength of mechanisms or binding of the security functions.

5.8.37 An example of a function that would not be suitable would be the use of a DAC function where there was a threat of someone attempting to access classified information for which they had insufficient clearance. This is because the DAC function has no way of telling what the classifications and clearances are of the object and subjects with which it operates.

5.8.38 Alternatively, the suitability argument could have been cast in terms of security objectives. This approach may be preferable in the case of a product or where the threat is expressed at a higher level of granularity, e.g. "there is a terrorist threat":

a) The end system access objective (S1) is met by the combination of the logon functions (for CM1 and CM4) and the access control function (CM2) (for the reasons given in Paragraph).

b) The confidentiality of information in transit (S2) is protected by the cryptographic mechanism (for CM3) (for the reasons given in Paragraph 5.8.34).

Binding Analysis

5.8.39 The ITSEC requires the sponsor:

a) to provide an analysis of all potential interrelationships between security enforcing functions and mechanisms;

b) to show that it is not possible to cause any security enforcing function or mechanism to conflict with or contradict the intent of other security enforcing functions or mechanisms.

5.8.40 Unlike suitability of functionality, **binding analysis** must take account of the composition of the system, i.e. the developer must consider *all potential interrelationships between security enforcing functions and mechanisms*.

5.8.41 In this example, end system access control is violated if RED data can be displayed on a BLUE terminal. Communications confidentiality is violated if the crypto keys are compromised, both terminal or host cryptos are bypassed and transmission is "in clear", or any other "useful" information is transmitted in-clear.

5.8.42 The sponsor's binding analysis shows that:

a) If the user (attacker) fails to logon to SWAN, for whatever reason, no useful information can be obtained.

b). If the user logs on to the SWAN successfully, the cryptographic devices located between the terminal and SWAN operate in bypass mode until a connection is established with a host computer. Therefore, SWAN authentication data will be sent over the SWAN in clear.

c) The user is only offered services (i) for which he is authorised and (ii) which satisfy the SWAN access control policy.

d) A crypto link is then established between the terminal and the host. The developer assumes that only good keys are used.

e) The user then logs on to the host. If this fails, then the process terminates: no useful data will have been displayed on the terminal and no other useful data will have been transmitted across the SWAN (other than the SWAN authentication data, see (b) above).

f) If the user is successful he/she may then transfer *encrypted* information between his/her terminal and the host.

5.8.43 There are three scenarios presented in the sponsor's Binding Analysis, as shown in figure 5.8.3.

Figure 5.8.3 Binding Analysis

Scenario	**Data displayed**	**Data on SWAN**
User fails to log onto the SWAN	None	None
User successfully logs on to the SWAN but fails to log on to a host	None	I&A information *in-clear*
User successfully logs on to the SWAN and to an *authorised* host service	BLUE data	I&A information *in-clear* encrypted data

5.8.44 Consequently the sponsor is able to show that:

a) The SWAN I&A, Access Control and Host I&A functions *do bind*, since for all scenarios, RED data is never displayed in clear.

b) The encryption devices, however, *do not entirely bind together*, since for some scenarios, SWAN authentication data is transmitted in clear.

5.8.45 The sponsor then argues that defeating CM1 is not itself sufficient to cause a violation of the security objectives because, although SWAN authentication data is transmitted "in clear", the cryptographic mechanisms (for CM3) and host logon (for CM4) still enforce the security policy.

5.8.46 The evaluators perform an independent check of the sponsor's Binding Analysis. The apparent lack of binding is noted but the evaluators do not assign a fail verdict at this point in the effectiveness assessment.

5.8.47 This binding failure merely presents a vulnerability which the evaluators must independently check to see whether it is exploitable in practice. At this stage it is not possible to assign a fail verdict unless it can be shown that this vulnerability (in construction) is exploitable. This cannot be determined until the evaluators have applied the vulnerability assessment criteria and performed penetration testing.

Sponsor's Vulnerability Analyses

5.8.48 In accordance with the ITSEC (Paragraphs 3.26 to 3.27 and 3.35 to 3.36) the sponsor provides the evaluators with a list of known vulnerabilities in the construction and operation of the TOE along with an analysis of the potential impact of each known vulnerability on the security of the TOE.

5.8.49 In this example, the sponsor has combined the list of known vulnerabilities in construction and the list of known vulnerabilities in operation and has performed a single vulnerability analysis.

5.8.50 Operational vulnerabilities are associated with physical and administrative procedures external to the TOE. They may provide the attacker with the opportunity and resources necessary to make use of a **construction vulnerability** or to make a direct attack. They may also provide the attacker with the security information (e.g. an user identifier and password) necessary to masquerade as an authorised user.

5.8.51 The ITSEC requires the sponsor to show that the vulnerability is not exploitable in practice, i.e. that:

a) each vulnerability is adequately covered by other, uncompromised, security mechanisms, or;

b) the vulnerability is irrelevant to the security target, will not exist in practice, or is countered by technical, personnel, procedural or physical countermeasures outside the TOE.

5.8.52 The construction and operational vulnerabilities known to the sponsor are given in figure 5.8.4. This list relates the vulnerabilities to threats (i.e. threat T1, the user could masquerade as another user when accessing the SWAN, perhaps as a result of eavesdropping SWAN I&A data) and the security objective which may be violated if the vulnerability is exploitable in practice.

5.8.53 In order to gain access to host data in violation of the security policy, an attacker must successfully effect an *attack scenario* through the four countermeasures (CM1..CM4). Each countermeasure along the attack scenario must be overcome by either direct attack (e.g. an attack based on the underlying algorithms, principles or properties of the countermeasure concerned) or by indirect means (e.g. by bypassing).

5.8.54 Figure 5.8.5 shows the sponsor's analysis of all the possible attack scenarios which would result in a violation of the security objectives, that is, to:

a) protect against unauthorised access to an end system (S1);

b) protect the confidentiality of the information in transit (S2).

5.8.55 The means by which the countermeasures CM1..CM4 may be defeated includes a manifestation of the corresponding threats T1..T4, which may be accomplished by indirect attacks on these countermeasures using those vulnerabilities identified above (e.g. V1, V2 are indirect attacks on CM1).

5.8.56 If an attacker has a valid account on the host then the attacker has a choice of using the SWAN authorised services menu normally or invoking V6. If the attacker does not have a valid account on the host then the attacker may only use V6 because the presence of the target host in the authorised services menu, under normal circumstances implies that he has a valid account on the host. V6 requires collusion with the SWAN security administrator but not the security administrator of the target host, necessitating the need to attack the host logon.

5.8.57 Although it is the developer's intention that the countermeasures should be overcome in the order CM1, CM2, CM3 and CM4 it is possible, due to the construction of the SWAN that other attack scenarios may exist that could lead to an **exploitable vulnerability**.

5.8.58 In this example, the sponsor's analysis shows no means of bypassing the cryptographic mechanism, the only identified vulnerability being deactivation by an accomplice. Therefore, for each attack scenario shown in figure 5.8.5, the attacker must defeat countermeasure CM3 in order to violate the security objectives S1 and S2.

5.8.59 CM3 is rated *medium* SoM against direct attack and meets the minimum claimed SoM for the SWAN. An attacker could attempt to take advantage of the construction vulnerability V4, but this clearly requires collusion with a bone fide user of the target host and is addressed by security measures outside of the TOE.

5.8.60 The sponsor's analysis therefore shows that the known vulnerabilities shown in figure 5.8.5 are adequately countered by the cryptographic devices (and measures outside of the TOE).

5.8.61 Since there is no apparent means, other than those listed above, to bypass any countermeasure in the attack scenario, it is clear that the sponsor's analysis has taken into account all combinations of known vulnerabilities. Similarly, based on their knowledge of the TOE, the evaluators are satisfied that the sponsor's analysis makes no unreasonable assumptions about the intended environment.

Figure 5.8.4 List of Known Construction and Operational Vulnerabilities

ID	Description	Threat	Security Objective
V1	**Attacker eavesdrops on SWAN I&A data**. The attacker eavesdrops on the SWAN network and obtains the SWAN identifier and password for a user of the target host (the host to be attacked). This was the vulnerability identified by the Binding of Functionality analysis above.	T1	S1
V2	**Break key on SWAN logon.** Pressing the break key during a SWAN logon causes the logon process to time out after 5 minutes, provided that the user does not type anything. A timeout message appears. Provided the user does not do anything for 10 minutes, the timeout message itself times out whereupon the authorised service menu is displayed for the last user who successfully logged onto the SWAN. That user could have been a user of the target host.	T1	S1
V3	**Unauthorised services available.** Provided the number of authorised services for the current user are less that those for the user who previously logged on the SWAN, the additional (unauthorised) services are still available even though not displayed.	T2	S1
V4	**Accomplice deactivates cryptos.** Pressing the break key twice in rapid succession whilst logging on to the host causes the TCU to enter bypass mode without signalling the NMC to break the session with the host. A subsequent depression of the break does, but due to a defect in the HCU, this causes all further transmissions between that HCU and any TCU are in clear. Since logon to the host is protected by the HCU, deactivation of the HCU is only possible by an authorised user of that host - hence the need for an accomplice.	T3	S2
V5	**Accomplice has captured host I&A data.** Use of a particular sequence of function keys permits a user to gain access to the HOST's password table. Passwords are encrypted, but possible to crack within a few days. It can only be performed by a bone fide user of the host as in the case of V4 above	T4	S1
V6	**Attacker has authorised service by collusion.** Authorised services are established for a user upon written request by the user and authorised by his/her line manager. The data is checked for consistency with other users and is rejected by the NMC if it fails to meet the SWAN's access control policy. The data is entered via a two-man rule. Even so, it is just possible, with collusion, to use the same system number, S#, and security level, SL, as those for the target host to identify a new system and use DAC controls to separate the two user communities of the two systems, but with the attacker being granted access to both.	T2	S1

Figure 5.8.5 Sponsor´s Analysis of the Attack Scenarios

Seq	Description	CM1(basic)	CM2(high)	CM3 (medium)	CM4 (basic)	Violates
1	Attacker defeats CM1, uses the SWAN authorisation menu normally, defeats CM3 and successfully logs on to the host	V1 or V2	use menu	V4	logon to host as auth. user	S1,S2
2	Attacker defeats CM1, uses the SWAN authorisation menu normally, and then defeats both CM3 and CM4	V1 or V2	use menu	V4	V5	S1,S2
3	Attacker defeats CM1..CM4 by using some combination of the above vulnerabilities	V1 or V2	V6 or V3	V4	V5	S1,S2
4	Attacker defeats CM1..CM3 and then performs a successful host logon	V1 or V2	V3	V4	logon to host as auth. user	S1,S2
5	Attacker successfully logs on to the SWAN and then defeats CM2..CM4	SWAN logon as auth. user	V6 or V3	V4	V5	S1,S2
6	Attacker successfully logs on to the SWAN, defeats CM2 and CM3 and then successfully logs on to the host	SWAN logon as auth. user	V3	V4	logon to host as auth. user	S1,S2
7	Attacker successfully logs on to the SWAN, selects an authorised service, and then defeats CM3 and CM4	SWAN logon as auth. user	select authorised service	V4	V5	S1,S2
8	Attacker successfully logs on to the SWAN, selects an authorised service, defeats CM3 and then successfully logs on to the host	SWAN logon as auth. user	select authorised service	V4	logon to host as auth. user	S2

Evaluators' Independent Vulnerability Analysis

5.8.62 The ITSEC requires the evaluators to perform an independent vulnerability analysis taking into account the listed and any other known vulnerabilities (both operational vulnerabilities and construction vulnerabilities) found during the evaluation.

5.8.63 Throughout this example it has been assumed that the correctness criteria have already been applied to the TOE. For the purposes of this example it is assumed that the application of the correctness criteria identified a potential vulnerability which, on analysis, was considered to be a construction vulnerability. This vulnerability was not already identified in the sponsor's list of known vulnerabilities. This vulnerability is identified in figure 5.8.6.

Figure 5.8.6 Construction Vulnerabilities Found During Correctness Assessment

ID	Description	Threat	Security Objective
V7	**User deactivates cryptos** As V4, but if the attacker is a user of the target host he is his own accomplice. This situation occurs for example if the attacker is a user of a system with MiC data with authorised access only from a cleared terminal room but the attacker wishes to access the system from his (uncleared) office.	T3	S2

5.8.64 The consequence of this construction vulnerability is that a user who successfully logs on to a RED host system is able to defeat countermeasure CM3 without the need for collusion. This vulnerability affects attack scenarios 1,4,6 and 8 in the evaluators' analysis of the attack sequences, which are now as is shown in figure 5.8.7. Of concern to the evaluators was that the sponsor's vulnerability analysis showed that:

a) CM1 and CM4 are rated as having a basic SoM and, in isolation, are not adequate (note, only the mechanisms for CM2 (*high*) and CM3 (*medium*) meet the claimed minimum SoM for the TOE (*medium*)).

b) the mechanism for CM3 is the only one in attack scenarios 1,2,7,8 which meets the claimed minimum SoM for the SWAN (attack scenario 8 relies entirely on CM3 to maintain the security policy).

c) If the mechanism for CM2 can be defeated then the mechanism for CM3 is the only other one in attack scenario 3,4,5, and 6 which meets the claimed minimum SoM for the SWAN.

Figure 5.8.7 Evaluators' Analysis of the Attack Scenarios

Seq	Description	CM1(basic)	CM2(high)	CM3 (medium)	CM4 (basic)	Violates
1`	Attacker defeats CM1, uses the SWAN authorisation menu normally, defeats CM3 and successfully logs on to the host	V1 or V2	select authorised service	V4 or V7	logon to host as auth. user	S1,S2
4`	Attacker defeats CM1..CM3 and then performs a successful host logon	V1 or V2	V3	V4 or V7	logon to host as auth. user	S1,S2
6`	Attacker successfully logs on to the SWAN, defeats CM2 and CM3 and then successfully logs on to the host	SWAN logon as auth. user	V3	V4 or V7	logon to host as auth. user	S1,S2
8`	Attacker successfully logs on to the SWAN, selects an authorised service, defeats CM3 and then successfully logs on to the host	SWAN logon as auth. user	select authorised service	V4 or V7	logon to host as auth. user	S2

5.8.65 Until the evaluators are able to prove whether or not this additional vulnerability in construction (or any of the above vulnerabilities) are exploitable in practice (through penetration testing), it is not possible to assign a final verdict against the vulnerability assessments (see *Evaluator Verdicts* section, part 4, chapter 4.4).

Strength of Mechanisms

5.8.66 Although the sponsor has rated all mechanisms (see Paragraph 5.8.25), the sponsors vulnerability analysis shows that the only critical mechanism is that of CM3 (this was also confirmed by the evaluators' independent vulnerability analysis).

5.8.67 Countermeasure CM3 is cryptographic and the assessment of the strength of its cryptographic mechanism is outside the scope of the ITSEC, as are the key management procedures. The evaluators can only check, by reference to the relevant National Authority, that the crypto mechanism satisfies the claimed minimum strength of mechanisms rating for the SWAN.

5.8.68 The evaluators need to ask the relevant National Authority for the context of on-line crypto-analysis (required for an attack on end system separation, because of the need to establish a communication path with the host) and for the context of off-line crypto-analysis (for eavesdropping) whether the SoM of the cryptos, including key management procedures, would be rated at least *medium*.

5.8.69 For the purpose of this example, it is asserted that the answer to both of these questions is affirmative and, since the critical mechanism is a crypto, no penetration testing with regards to direct attack or operational vulnerability is undertaken by the evaluators. Consequently the evaluators are able to assign a pass verdict against the strength of mechanisms criterion.

5.8.70 In this example there is just one critical mechanism, common to both security objectives. In other cases, where there are multiple security objectives, the critical mechanism for each objective could be different.

5.8.71 Also, on-line cryptographic attack is very difficult, if not impossible, without circumventing the key management procedures in order that "bad" keys are used. In an off-line attack, as would be the case for attack scenario 8', it might be possible to deduce the key from analysis of the cipher text. In this example, the SoM of the algorithm and key management procedures are sufficiently strong to prevent this from happening.

5.8.72 The sponsor's Strength of Mechanisms analysis provides a detailed justification for describing the SWAN access control mechanism (CM2) as a hard mechanism, i.e. it is not susceptible to direct attack (see part 6, annex 6.C). This is reflected in the Sponsor's analysis by assigning this mechanism a *high* strength of mechanism.

Ease of Use

5.8.73 This aspect of effectiveness investigates whether the TOE can be configured or used in a manner which is insecure but which an administrator or end-user of the TOE would reasonably believe to be secure.

5.8.74 The sponsor's ease of use analysis needs to identify possible modes of operation of the TOE, including operation following failure or operational error, their consequences and implications for maintaining secure operation. It must also show:

a) that any human or other error in operation that deactivates or disables security enforcing functions or mechanisms will easily be detectable;

b) that if it is possible to configure or cause the TOE to be used in a way which is insecure (i.e. the security enforcing functions and mechanisms of the TOE do not satisfy the security target), when an end-user or administrator of the TOE would reasonably believe it to be secure, then that fact is also easily detectable.

5.8.75 The known insecure states of the TOE are identified by the attack scenario. Their mere existence indicates that it *is* possible to use or configure *(attacker has authorised service by collusion* is a configuration problem*)* the TOE in an insecure manner. The question is therefore, in such a case would an administrator or end-user reasonably believe the TOE to be secure.

5.8.76 The sponsor asserts that the behaviour of the TOE following failure or operational error, including their consequences and implications for maintaining secure operation, have been dealt with already - otherwise the sponsor's lists of vulnerabilities would be incomplete. In other words, if some new vulnerability was introduced (e.g. electrical failure of the cryptos) at this stage it would be necessary to re-visit the vulnerability assessment criteria.

5.8.77 Given the previous analyses in this example, this criterion simply addresses whether it is possible to detect whether the critical mechanisms of the TOE have failed. If a critical mechanism fails then the TOE is in, or is in danger of entering, an insecure state. The ITSEC criterion requires the TOE to merely detect this.

5.8.78 The sponsor notes that each crypto box has a lamp which is illuminated when that unit operates in cipher mode and is extinguished when it is in clear mode. The evaluators know that these boxes function correctly (for the purposes of this example it is assumed that the correctness criteria have been applied successfully). The lamps should be permanently on for all active host HCUs, providing a clear indication that both end system separation and communications confidentiality are satisfied.

5.8.79 It should be noted, however, that this analysis may become more complex if other functions (e.g. accountability functions) are included in the security target.

Penetration Testing

5.8.80 At this point in the evaluation of the SWAN, the evaluators have completed the all correctness activities and have assigned a final *pass* verdict against all of the correctness of the SWAN. However, as a result of a potential vulnerability identified in the correctness assessment, the evaluators highlighted a vulnerability in construction which had not been identified by the sponsor and therefore was not included in the sponsor's vulnerability analysis.

5.8.81 As discussed in part 4, chapter 4.4, the evaluators are unable to assign a final effectiveness verdict until penetration testing has been completed. The objective of penetration testing (as defined by the ITSEC) is to confirm or disprove whether the known vulnerabilities in the construction or operation of the SWAN are actually exploitable in practice.

5.8.82 In this example, the evaluators' penetration testing of the SWAN confirms that if an attacker is the user of a host, then the attacker can disable the crypto devices (V7) without the need for specialist knowledge or tools (the evaluators achieved this unaided and within minutes). Referring back to the evaluators' independent vulnerability analysis, figure 5.8.7 shows that attack scenario 1', 4', 6' and 8' are all affected by this vulnerability.

5.8.83 The cryptographic mechanism is the only critical mechanism in the SWAN. For attack scenarios 1' and 8', if the cryptographic mechanism fails then security objective S2 is immediately compromised and security objective S1 is then defended only by *basic* mechanisms which do not meet the claimed rating of the Minimum Strength of Mechanisms of the SWAN (*medium*).

5.8.84 However, attack scenarios 4' and 6' also require the attacker to defeat CM2 (rated *high*) in order to violate security objective S2, but the results of the evaluators' penetration testing of the SWAN shows that CM2 could be defeated by an attacker working unaided and within a few days (as a result of a construction vulnerability V3).

5.8.85 Therefore, vulnerability V7 is exploitable in practice and the evaluators assign a *fail* verdict against the construction vulnerability analysis, and therefore a final *fail* verdict is assigned against the effectiveness of the TOE.

Chapter 5.9. Example 8, Examine the Developer's Security (E2 and E4)

Introduction

5.9.1 This example presents two sub-examples (8(a) and 8(b)) each of which addresses one development environment aspect at different evaluation levels.

Example 8(a) - Examine the Developer's Security (E2)

Introduction

5.9.2 This sub-example covers the development environment aspect 3 - Developer's Security actions. The primary objective of this example is to illustrate how the Developer's Security can be examined. Physical and procedural security measures were used to protect the development environment.

ITSEC Requirements for Contents and Presentation

5.9.3 E2.21 The document on the security of the development environment shall state the intended protection for the integrity of the TOE and the confidentiality of the associated documents. Physical, procedural, personnel and other security measures used by the developer shall be stated.

ITSEC Requirements for Evidence

5.9.4 E2.22 The information on the security of the development environment shall state how the integrity of the TOE and the confidentiality of the associated documentation are maintained.

ITSEC Evaluator Actions

5.9.5 E2.23 Check that the documented procedures are being applied. Check that the information provided meets all requirements for content and presentation and evidence. Search for errors in the procedures.

Relevant Evaluation Deliverables

5.9.6 The input to this work is the information on the security of the development environment provided by the sponsor and the security target of the product or system containing the assumed or actual threats.

Work Performed

5.9.7 The information on the security measures was *stated* in the developer's security documentation. The documentation was examined by the evaluators (by reading and understanding it). In particular, the evaluators checked that:

a) the physical security measures were appropriate to protect the development environment from deliberate attack;

b) the procedural security measures were adequate to protect the integrity of the TOE and to maintain the confidentiality of the associated documentation.

5.9.8 It was made possible for the evaluators to visit the development site and to confirm that the security measures stated by the sponsor were being applied by:

a) assessment of other delivered documentation for conformance to the procedures of the security measures;

b) interviewing development staff to ascertain whether they were aware of the procedures and followed them in practice.

5.9.9 In order to further check that the documented procedures were being applied, the evaluators then:

a) checked the application of the physical security measures;

b) checked the application of the procedural security measures.

Example 8(b) - Examine the Developer's Security (E4)

Introduction

5.9.10 This sub-example covers the *development environment aspect 3 - developer's security actions.* The primary objective of this example is to illustrate how the Developer's Security can be examined. Physical, procedural and technical security measures were used to protect the development environment.

ITSEC Requirements for Contents and Presentation

5.9.11 E4.21 The document on the security of the development environment shall describe the intended protection for the integrity of the TOE and the confidentiality of the associated documents. Physical, procedural, personnel and other security measures used by the developers shall be described.

ITSEC Requirements for Evidence

5.9.12 E4.22 The information on the security of the development environment shall describe how the integrity of the TOE and the confidentiality of the associated documentation are maintained.

ITSEC Evaluator Actions

5.9.13 E4.23 Check that the documented procedures are being applied. Check that the information provided meets all requirements for content and presentation and evidence. Search for errors in the procedures.

Relevant Evaluation Deliverables

5.9.14 The input to this work is the information on the security of the development environment provided by the sponsor and the security target of the product or system containing the assumed or actual threats.

Work Performed

5.9.15 The information on the security measures was *described* in the developer's security documentation. The documentation was examined by the evaluators (by reading and understanding it in detail). In particular, the evaluators checked that:

a) the physical security measure were appropriate to protect the development environment from deliberate attack;

b) the procedural security measures were adequate to protect the integrity of the TOE and to maintain the confidentiality of the associated documentation;

c) the technical security measures were adequate to protect the integrity of the TOE and to maintain the confidentiality of the associated documentation.

5.9.16 During the pre-evaluation phase a problem within the configuration control system was noted. Every member of the development team could modify the TOE source code produced by any other development team member in an unauthorised manner. The problem was solved by activating the access control features of the configuration control system, so that every developer could only modify his own source code.

5.9.17 It was made possible for the evaluators to visit the development site and to confirm that the security measures described by the sponsor were being applied by:

a) assessment of other delivered documentation for conformance to the procedures of the security measures;

b) interviewing development staff to ascertain whether they were aware of the procedures and followed them in practice.

5.9.18 In order to further check that the documented procedures were being applied, the evaluators then:

a) checked the physical security measures by testing them. The evaluators checked that there was no way to circumvent the procedures used;

b) checked the procedural security measures by testing them. The evaluators checked the suitability of the procedures used;

c) check the technical security measures by testing them. The evaluators checked the suitability of the used procedures in accordance with the tool based configuration control system.

Part 6 Guidance to Other Parties

Contents

Figures

Chapter 6.1 Introduction

Objective of this Part

6.1.1 The objective of this part is to give guidance to sponsors, developers, vendors, users and system accreditors involved in Information Technology (IT) security. This guidance is intended to enable these parties to use the ITSEM in the most effective way and to enable them to gain a better understanding of the evaluation and certification process.

6.1.2 The effective use of IT is essential for increasing business prosperity; dependence on IT is increasing, as is the diversity of usage within all sectors of commerce and industry. However, there are potential risks associated with the use of IT. So, it is important that security is considered, preferably from the outset, and appropriate safeguards deployed. The consequences of not doing so can be dramatic, including loss of **assets**, damage to business reputation, inability to meet legal or market requirements, or even business failure.

6.1.3 Users are not always able to make a detailed analysis of the security provided by a product or a system, and they wish to base their trust in its assurance level.

6.1.4 To provide attractive products or systems, it is necessary to introduce new features, but these should be offered in the best "time to market" and with a controlled production cost.

6.1.5 The above paragraphs lead to the following questions:

a) Is it useful to provide security without assurance?

b). Is it possible to have assurance without evaluation?

c) Can evaluation be done without excessive extra cost?

6.1.6 Security should be considered as an inherent quality of any product or system and the evaluation process is the means by which the assurance level offered by the security of a product or system can be determined.

Relationship of this Part to the other Parts of ITSEM

6.1.7 This part is aimed at all the parties involved in IT security, as defined in part 1 of the ITSEM, and describes the roles and activities of these parties within the evaluation process.

6.1.8 Specific guidance to parties involved in the certification process (ITSEF, sponsor, and **certification body**) is provided in part 2 of the ITSEM, and in the **national scheme** documentation.

6.1.9 Specific guidance to evaluators (ITSEFs) involved in the evaluation process is provided in part 4 of the ITSEM.

6.1.10 This part provides guidance for aspects of IT security not addressed by the other parts of the ITSEM:

a) Preparation for evaluation: guidance is provided to ensure that those parties involved in the evaluation and certification processes are adequately prepared for an efficient evaluation.

b) Before or during the evaluation: guidance is provided on the development process.

c) After the evaluation: guidance is provided on the use of evaluation outputs.

d) After the certification: guidance is provided on the use of the **certificate/certification report.**

e) After the evaluation and certification: guidance is provided on the modification of an evaluated system or product.

Structure and Summary of this Part

6.1.11 The structure of this part is as a set of chapters and annexes, with these introductory remarks forming chapter 6.1.

6.1.12 The parties involved in IT Security and their responsibilities are described in chapter 6.2.

6.1.13 Guidance to security providers (i.e. sponsors, developers and vendors) is given in chapter 6.3.

6.1.14 Guidance to security procurers (i.e. users and system accreditors) is given in chapter 6.4.

6.1.15 Annex 6.A provides guidance to sponsors and developers on the supply of evaluation **deliverables** to the evaluators.

6.1.16 Annex 6.B is intended for sponsors and system accreditors and provides an example of the derivation of a security target.

6.1.17 Annex 6.C provides guidance on mechanisms and effectiveness.

6.1.18 Annex 6.D is intended for sponsors and system accreditors and describes **impact analysis** as a means of determining the consequences to certification of changes made to an evaluated system or product.

6.1.19 Annex 6.E provides general guidance to evaluation tool developers.

6.1.20 Annex 6.F is intended for sponsors, system integrators and system accreditors, who are concerned with the composition of previously evaluated TOEs.

Chapter 6.2. Parties Involved in IT Security

Introduction

6.2.1 The parties involved in IT security (see part 1 of the ITSEM) are as follows:

a) Sponsors, who request an evaluation, define the security target of a product or system to be evaluated, bear the cost of the evaluation, and receive the certificate/certification report.

b) Developers (including system integrators), who produce the product or system to be evaluated and provide the requested deliverables for the evaluation.

c) ITSEFs, which evaluate the product or system.

d) National certification bodies, which monitor the evaluation process and issue the certificates/certification reports.

e) Vendors, who sell and distribute evaluated products.

f) Users, who make use of an evaluated product or system to protect their assets.

g) System accreditors, who are responsible for the security of an evaluated system.

6.2.2 It is possible for a single party to fulfil more than one role, becoming, for example, sponsor and developer, or sponsor and vendor, or user and system accreditor, etc.

6.2.3 Specific guidance to ITSEFs and national certification bodies is outside the scope of this part.

Responsibilities of the Parties Involved

6.2.4 The objectives of the parties involved could be classified as:

a) to ensure adequate security is provided by a TOE;

b) to reduce or control the costs of providing that security;

c) to provide the required security within an acceptable timeframe.

6.2.5 In many cases it is necessary to find a compromise between these objectives.

6.2.6 The sponsor is responsible for:

a) definition of the security target;

b) definition of the TOE;

c) supply of the deliverables requested for the evaluation;

d) the use made of the certificate/certification report;

e) maintenance of the evaluation rating.

6.2.7 The developer is responsible for:

a) specification of the TOE;

b) product or system development;

c) producing the deliverables as requested for the evaluation;

d) maintenance of the product or system;

e) protection of his know how and proprietary information.

6.2.8 The vendor is responsible for:

a) product distribution;

b) product advertising;

c) providing advice;

d) product installation.

6.2.9 The user is responsible for:

a) product or system selection;

b) product or system start up;

c) product or system use;

d) product or system configuration.

6.2.10 The system accreditor is responsible for:

a) specification of the system security policy;

b) specification of the system modification rules;

c) calculation of the required assurance level;

d) approving a system's operational use.

6.2.11 This list is not definitive because different organisations are likely to assign responsibilities differently.

6.2.12 An ITSEF can act as an advisor in the specification or realisation of the TOE, but would be unable to provide any advice that would affect its independence (see part 4, chapter 4.2).

Chapter 6.3 Guidance for Sponsors, Developers and Vendors (Security Providers)

Introduction

6.3.1 Security providers are those who provide input to the evaluation process (i.e. sponsors and developers) and those who provide security services (i.e. vendors). This chapter covers the following topics:

a) definition of the security target (relevant to sponsors);

b) initiating product evaluations (relevant to sponsors and vendors);

c) supplying and managing deliverables (relevant to sponsors and developers);

d) the development process (relevant to developers);

e) specialised development techniques (relevant to developers);

f) using ETRs and certificates/certification reports (relevant to sponsors);

g) certificate maintenance (relevant to sponsors and developers);

h) selling certified products (relevant to vendors);

i) installing and configuring products (relevant to vendors);

j) integrating products (relevant to vendors and developers);

k) providing advice (relevant to vendors).

Definition of the Security Target

6.3.2 It is the sponsor's responsibility to provide the security target for a TOE. The objectives of a security target are as follows:

a) to provide a specification of a TOE's security functionality;

b) to relate a TOE to the environment in which it is intended to operate;

c) to provide the basis of the evaluation.

6.3.3 The intended audience for a security target may therefore include:

a) the developer of the TOE - the security target defines the security requirements of the TOE;

b) the evaluators - the security target provides the baseline against which the TOE is evaluated;

c) the vendor or the user of a TOE - the security target specifies the security objectives of the TOE to those responsible for managing, purchasing, installing, configuring and operating the TOE.

6.3.4 As stated in ITSEC (Paragraphs 2.4 - 2.26 and 4.11) the required content of the security target is determined by whether the TOE is a system or a product. The content can be summarised as follows:

a) either a system security policy, or a product rationale;

b) a specification of the required security enforcing functions;

c) a definition of required security mechanisms (optional);

d) the claimed rating of the minimum strength of mechanisms;

e) the target evaluation level.

6.3.5 The security target is the basis for the evaluation and is itself subject to evaluation.

6.3.6 Producing a security target for a TOE which meets the ITSEC criteria requires a thorough application of a methodical approach. In particular, the security target should be defined using a top-down approach considering, in turn:

a) limitation of the domain: risk analysis;

b) operational specifications: security policy;

c) functional specifications: the security enforcing functions;

d) implementation specifications: the required mechanisms and the minimum strength of mechanisms;

e) evaluation specifications: the target evaluation level.

6.3.7 Further guidance on the content of a security target and a top-down approach to producing it is given in annex 6.B.

Initiating Product Evaluations

6.3.8 It is often more cost-effective to use off-the-shelf solutions to meet general requirements. This can be true for security requirements as well as for any other requirements.

6.3.9 A product can be considered from the security point of view as being one of the following:

a) a security product, designed with a specific security purpose as its only or primary purpose (e.g. a product which implements identification and **authentication** on a desktop PC);

b) a secure product, aiming to provide a specific level of security as a complement to its much wider functionality (e.g. an operating system).

6.3.10 The decision to develop and market a security product or a secure product may be based on factors which include:

a) threats perceived by users to their assets (e.g. virus attacks);

b) national or international legal requirements (e.g. the US Computer Security Act);

c) national or international standards (e.g. provision of security in X.400 or X.500);

d) a market niche (e.g. access control devices for personal computers).

6.3.11 The business environment will always have an overriding effect on the decision. The sponsor will have to consider a number of questions relevant to the commercial viability of the product. These questions might include:

a) Who are the potential customers?

b) Why is security an issue to these potential customers?

c) What level of security (in terms of functionality and assurance) is required by these potential customers?

6.3.12 The requirements and implications of product certification will also be addressed:

a) Should evaluation and certification under a recognised scheme be sought?

b) What is the commercial and legal impact of such a decision (e.g. export control)?

6.3.13 Given certain assumptions concerning the above points, the sponsor should build a business plan for his product, including a consideration of the likely competitors in this area.

Supplying and Managing Deliverables

6.3.14 The sponsor is responsible for supplying the deliverables to the evaluators during the evaluation process.

6.3.15 The term *deliverabl*e is used to refer to any item (including the TOE itself) that is required to be made available to the evaluators for evaluation purposes. This includes intangible items, such as support to the evaluators (e.g. training where necessary) and access to computers.

6.3.16 The purpose of deliverables is to enable the evaluators to evaluate the TOE. Different types of deliverables satisfy this purpose in different ways, such as:

a) deliverables which provide evidence of effectiveness or correctness, e.g. an informal description of correspondence between source code and detailed design;

b) deliverables which allow the evaluators to establish additional evidence of effectiveness or correctness, e.g. access to the developed TOE;

c) deliverables which improve the overall efficiency of the evaluators' work, e.g. technical support from the developer.

6.3.17 Detailed guidance to sponsors and developers on the contents and management of deliverables is provided in annex 6.A.

6.3.18 The sponsor should ensure that the arrangements with the developer are both:

a) sufficiently definitive to ensure that the evaluators receive the required deliverables;

b) sufficiently binding to ensure that inadequate deliverables result in a contractual shortfall.

6.3.19 It is the sponsor's responsibility to supply the evaluators with any required deliverables produced by subcontractors, or associated with third-party products (e.g. source code).

6.3.20 Failure to provide the required deliverables within reasonable timescales, or failure to provide deliverables of adequate quality, may result in the evaluation being suspended until acceptable deliverables are made available, since further work on the evaluation may not be possible.

6.3.21 The developer has to provide all of the expected deliverables by the due dates agreed at the start of the evaluation. To fulfil this obligation, the developer should:

a) confirm the correspondence between the deliverables list and his development plan;

b) confirm the correspondence between the deliverables list and the outputs of his development process;

c) confirm the correspondence between the expected level of information and his development methods.

6.3.22 On occasion, the developer may accede to the evaluation and the supply of deliverables to the ITSEF, but may wish to limit the sponsor's access to proprietary information. The developer, at the appropriate time, should ensure that the nature and extent of the proprietary information is defined and should establish basic rules for protecting it.

6.3.23 Before an evaluation the sponsor should, in accordance with national legal regulations:

a) establish all necessary legal rights to the TOE and other deliverables, for the purposes of evaluation and to grant rights to (indemnify) the ITSEF and certification body in this respect;

b) (where appropriate) obtain the written consent of the developer to any specific arrangements to limit access to proprietary information.

6.3.24 As a consequence of the decision to have a product or a system evaluated, the developer should agree to accept its responsibilities in the evaluation process.

The Development Process

6.3.25 Developers are expected to provide deliverables as evidence that the targeted assurance level has been achieved (see annex 6.A). This evidence should be prepared as part of the development process or after the development if evaluation was not the original objective.

6.3.26 The development process is assumed by the ITSEC to consists of four phases:

a) The requirement phase:

For a system, this phase is the sponsor's responsibility (though often the developer of a product will also be the sponsor for an evaluation). It is important for the developer that, during this phase, the set of security requirements and their rationale are clearly defined and analysed to determine the strengths and weaknesses of the proposed product or system.

b) The architecture phase:

In this phase, the security requirements are used to develop a security architecture and to determine a set of security functions. Specific attention should be given to the separation of the security enforcing and security relevant functions from the other functions.

c) The detailed design phase:

This phase is a refinement of the architecture phase where the functionality of each component becomes apparent. Specific attention should be given to the strengths and weaknesses of candidate programming languages in the context of required security functions and mechanisms.

d) The implementation phase:

During this phase the developer implements the functions which provide the security features described in the design phase. Specific attention should be given to the application of the development rules and inspections or walk-throughs should be part of the development methodology.

Also, the developer executes a pre-defined test plan. Specific attention should be given to the completeness aspect of the test plan, and to the record of tests performed and the corresponding results, to be provided as deliverables for the evaluation.

6.3.27 The following general guidance is applicable to all the above phases as an aid to fulfilling the ITSEC criteria:

a) Developers should follow a structured approach to aid the production of code that is easy to read, maintain, and trace through the levels of refinement.

b) By analysing design information and source code of a security-relevant component, an attacker may be able to discover a way to breach a security objective. Therefore, developers should protect their proprietary information.

c) Developers are advised to give programmers a direct responsibility for the programs they develop. This will assist the understanding of security requirements within the development.

d) Developers should adopt a peer review process as part of the process of identifying potential security problems and malfunctions.

Specialised Development Techniques

Introduction

6.3.28 This section provides guidance to developers on specialised development techniques relevant to the higher assurance levels.

Tool Based Configuration Control Systems

6.3.29 At higher evaluation levels, developers are required to use a tool-based configuration control system. This subsection provides advice on selecting and developing such systems.

6.3.30 The configuration management system should ensure that there is a clear, complete and accurate **representation** of the TOE at all stages in its life cycle. This representation should reflect all changes that have been made to the configuration.

6.3.31 A tool based configuration control system should enforce a clearly described configuration management policy and should encompass:

a) the traceability of every modification of the TOE to an approved change request;

b) the traceability from effect to cause of any malfunction in the TOE due to a change;

c) analysis of the effect of changes on unchanged components;

d) the definition of responsibilities for change control;

e) the control of access to TOE software modules during their development;

f) the synchronisation of implementing TOE changes and updating TOE documentation;

g) the generation of any previous version of the TOE;

h) the auditing of implemented control procedures;

i) the auditing of the TOE status accounting procedures.

6.3.32 It is necessary to provide confidence that the TOE has been implemented in a controlled manner. All alterations to the TOE should be authorised and controlled to ensure that they do not adversely affect the ability of the security enforcing and security relevant components to enforce the system security policy or product rationale. The use of digital signatures may be helpful here.

Formal Methods

6.3.33 The use of formal methods, mandated by ITSEC at higher evaluation levels, sometimes causes problems for developers because of its novelty. This subsection provides guidance on selection of formal techniques and tools.

6.3.34 *Underlying Formal Specification and Description Techniques*: at level E6, ITSEC requires a formal description of the architecture of the TOE and a formal specification of the security enforcing functions.

6.3.35 Following ITSEC E6 requirements, formal comparison can be made between the formal description of the architecture and the underlying formally specified model of security. This comparison is not always easy: formal techniques are currently employed mainly for describing and proving static properties of TOEs. In this case, a combination of formal and informal techniques (as stated in ITSEC Paragraph E.6.6) becomes necessary.

6.3.36 Another comparison can be conducted between the formal specification of the security enforcing functions and their realisation in the TOE. A precise formal specification of the functionality of the TOE involves the use of a mathematical notation and is therefore abstract. It is a functional or semantic definition of what a system does, without stating how this should be accomplished. Given a formal specification of the functionality of the TOE, properties of the TOE can be formally stated and proved. It is also a precise standard for the implementation.

6.3.37 A formal style of specification is written in a formal notation based upon well-established mathematical concepts (ITSEC Paragraph 2.76). Most formal notations use the constructs of mathematical logic (predicate calculus and more recently modal logic) and set theory.

6.3.38 There are three complementary techniques or methods for formal description. Operational definitions use an abstract interpreter to define the TOE. These are the least abstract, and resemble implementations. Denotational descriptions map the TOE directly to its meaning. Axiomatic or equational definitions describe properties of the TOE.

6.3.39 It is recommended that developers select formal specification and description techniques based on the following considerations:

a) Levels: to allow the system designers or user to look at the formal description in as much or as little detail as desired, the description should be split into levels ranging from the top level flow of control to the details of each operation.

b) Modular: all but the top level of the formal description should be modular. This will enable the design of each operation to be considered in isolation.

c) Concise: the notation should allow the necessary concepts to be expressed in a concise manner. A notation which is clumsy or verbose will unnecessarily lengthen the description.

d) Understandable: the formal specification notation should be easy to understand.

e) Abstract: the formal description should not dictate any issues which need not be resolved until the implementation stage. Although the top level flow of control is vital to the design of the system, it is often the case that, at lower levels, the ordering of certain events is immaterial.

f) Sound: to allow formal proofs of correctness to be carried out, the description technique should have a sound mathematical basis.

6.3.40 *Formal specification tools* : briefly these are defined as tools which implement - and techniques which use - mathematical logic. These tools and techniques aim to provide conclusive proof that a TOE strictly satisfies its specification. Formal methods supported by a tool shall, more precisely, be defined by means of:

a) formally specified syntax and semantics for notations used;

b) algorithms to manipulate formulas of the languages;

c) a set of proof rules by which correctness (completeness and lack of ambiguity) of the specification may be inferred;

d) a framework for refining a specification into a concrete implementation.

6.3.41 *Expressiveness of Formal Specification Languages* employed by a tool shall be sufficient to formally describe the security policy and the components of an IT system enforcing that policy, eg. in terms of invariant predicates. For the formal specification language there shall be concepts to structure the design specification in hierarchically ordered specification levels to refine a design specification from top-level TOE specification down to low-level program specifications.

Using ETRs and Certificates/Certification Reports

6.3.42 In some national schemes certificates/certification reports are official statements by a governmental organisation and therefore subject to the rules for those official publications. Users of ETRs and certificates/certification reports should contact the appropriate national body listed in part 2 of the ITSEM.

6.3.43 It is the responsibility of the sponsor to relinquish his rights to any results of the evaluation that would compromise the developer's own proprietary information. In the case of failure of the evaluation, the sponsor should not use this result against the developer's interest.

6.3.44 At the end of an evaluation, the ETR is provided to the certification body. In the evaluation and certification process, the ETR is an interim document and does not represent the final decision of this process.

6.3.45 The ETR is released to the sponsor. It is released in confidence, without prejudice to the official certificate/certification report and on the understanding that it will be restricted to a limited circle of the sponsor's staff and should not be distributed to other parties without the agreement of the certification body. The ETR should be marked *Evaluation-in-Confidence.*

6.3.46 Where the sponsor has a concern regarding any statements in the ETR or certificate/certification report, he may discuss his concerns with the ITSEF or certification body, as appropriate.

6.3.47 The certification body reviews the ETR to determine the extent to which the security target is met by the TOE and to ensure that the ITSEF has conducted the evaluation in conformance with the ITSEM requirements; it is then able to confirm the claimed evaluation level. Its conclusions are recorded in the certificate/certification report.

6.3.48 The use of evaluation results and certificates/certification reports should be constrained by specific requirements of the national scheme.

Maintenance of Certificates/Certification Reports

6.3.49 A certificate/certification report only applies to the release/version of the TOE that was evaluated; any changes to a certified TOE will fall under the procedures established for re-evaluation (detailed guidance is provided in annex 6.D).

6.3.50 The sponsor may only market a product as a certified product on the basis of a valid certificate/certification report and shall ensure that configuration management procedures, appropriate to the evaluation level, are in place to prevent unauthorised modifications. The evaluators may be required to archive the evaluation material to allow **re-evaluation**.

6.3.51 If a TOE or its operational or development environment is subsequently changed, it is the responsibility of the sponsor to classify the change type and determine the consequences for the certificate/certification report.

6.3.52 The type of change determines whether the sponsor should notify the certification body of the change. It may also be necessary for the sponsor to arrange for a re-evaluation.

6.3.53 The developers involved in the maintenance process should consider establishing a dedicated security team to perform an impact analysis of all changes proposed or implemented.

6.3.54 The maintenance process may be aided by the individual responsibility assignment policy followed during development (see 6.3.27.c) and may include a review process dedicated to the preparation of the information required for the re-evaluation of the TOE, including:

a) a summary of changes since the previous evaluated release;

b) a description of all security relevant changes and the security analysis of those changes.

6.3.55 Sponsors and developers are encouraged to consider re-evaluation and certificate maintainance during TOE development and preparation for the initial evaluation.

Selling Certified Products

6.3.56 Sponsors, developers and vendors may be interested in selling certified products.

6.3.57 Those selling certified products have the following duties:

a) to provide the product certificate/certification report when requested by potential users;

b) not to make misleading claims about the product (e.g. claiming a product is certified when it is not or exaggerating the benefits of the product);

c) to report known problems in certified products to potential users;

d) if a **vulnerability** is found in a certified product, to inform existing users of it;

e) when a certified product changes, not to claim the new product as certified until the certificate/certification report has been upgraded.

6.3.58 The main document of interest to vendors while selling products is the security target.

Installing and Configuring Products

6.3.59 Installation and configuration is generally done by developers, vendors or (for simple products) users.

6.3.60 Those installing and configuring products should:

a) follow the product's delivery instructions accurately;

b) select configuration options in accordance with the product's configuration documentation, and record what was done so that the product configuration will be known thereafter;

c) follow the appropriate procedure for checking the authenticity of the TOE, and address any discrepancies found.

6.3.61 At this stage the *operational environment* documentation will be of most use to the vendor.

Integrating Products

6.3.62 It is frequently the case that a number of evaluated products will need to be integrated together into a combined product or system. This is often an issue for products.

6.3.63 If the developer wishes, he can produce a new security target for the integrated product or system and can arrange for an evaluation against the new security target. In this case, the **re-use** guidance in part 4, chapters 4.3 and 4.6, applies. After successful certification, the vendor can claim that the integrated product or system has been certified against the new security target.

6.3.64 Alternatively, the vendor can simply check that the integrated product or system satisfies all the assumptions in the security targets of all the separate products, without arranging for an evaluation. In this case, the vendor can claim that each product has been certified against its security target, but he can make no claim about the security target of the integrated system or product. In particular, he can make no claims about how well the certified products will work together.

6.3.65 Annex 6.F provides a simple model for the composition of two previously evaluated components. This will be of interest to those concerned with system integration.

Providing Advice

6.3.66 Users who are considering purchasing evaluated products will often request advice from developers, vendors or ITSEFs.

6.3.67 Those providing advice have the following duties:

a) To provide impartial advice, that is, the advice given should be in the user's best interests; any interest the advisor has in a particular product should be explained to the user.

b) Not to provide advice outside the advisor's field of competence.

Chapter 6.4. Guidance for Security Procurers

Introduction

Background

6.4.1 This chapter gives guidance to security procurers, that is, sponsors, system accreditors and users of evaluated systems and products. This chapter covers the following topics:

a) security evaluation (a basic introduction of interest to users);

b) users and evaluation (of interest to users);

c) requirements definition (of interest to system accreditors);

d) system acceptance (of interest to system accreditors);

e) accreditation maintenance (of interest to system accreditors).

6.4.2 This chapter does not aim to provide a general introduction to security concepts; these are covered in many other publications [GASSER]. It is merely concerned with explaining the meaning of security evaluation and its consequences for users and system accreditors.

Users

6.4.3 The following types of users exist:

a) end-users who make use of an IT system in order to perform their normal work;

b) operators, responsible for startup, closedown, backups and other routine aspects of system control;

c) administrators, responsible for creating userIDs, system configuration, assigning file permissions and similar high-level control functions.

6.4.4 These roles will involve varying degrees of influence on the security of an IT system, varying between no influence at all, and being critical to the maintenance of security.

System Accreditors

6.4.5 A system accreditor is an individual, or an organisation, that is responsible for the security of a system, including its physical, personnel and procedural security features as well as those technical features provided by an IT system.

6.4.6 System accreditors can include:

a) the owner of data to be held on an IT system, who may need assurance that it is secure;

b) a departmental security officer, with responsibility for all IT security within part of a large organisation;

c) a national organisation which is responsible for ensuring that information that is important to national security is protected.

6.4.7 When assessing a system's security, a system accreditor will typically base the assessment on an organisational policy for security, which may be defined for a department or organisation, or in some cases just for the system of interest. This security policy should identify any security rules or regulations which apply to the system, including any non-IT requirements which should be enforced.

6.4.8 In order to establish confidence in a system's security, a system accreditor will be performing a process rather like a high-level evaluation, verifying that the combination of IT, physical, personnel and procedural measures effectively enforces the security policy which applies to the system.

6.4.9 The detailed technical evaluation of the IT components of a system will typically be performed by an ITSEF. A system accreditor will need to understand the process of IT system evaluation and certification to a level that allows the results of evaluation to be used in the accreditation activity.

6.4.10 The involvement of a system accreditor in the lifecycle of a secure system occurs primarily in three phases:

a) during the initial definition of requirements;

b) when approval is needed for a system to become operational;

c) whenever the system is changed or upgraded.

Security Evaluation

6.4.11 It is impossible to produce practical IT systems which are absolutely secure. This is because of the complexity of IT systems, and the variety of threats which they have to counter.

6.4.12 It is possible, however, to provide some confidence in the security of a computer system. The favoured approach is for an independent body (called an IT Security Evaluation Facility, or ITSEF) to examine the system design and documentation in detail to search for security vulnerabilities. This examination is called a security evaluation. A system passes its evaluation if it is found to be free from exploitable security vulnerabilities; otherwise it fails.

6.4.13 If a system has passed a security evaluation, it is likely that it will provide some degree of security but it cannot be considered absolutely secure, for the following reasons:

a) vulnerabilities may exist which have not been discovered by the evaluators, due to the level of information available to the evaluators;

b) the system may be used, operated, managed or configured insecurely;

c) some of the threats in the environment may not have been included in the security target.

6.4.14 Therefore, an evaluated system should be seen as having a role in maintaining an organisation's security, but it does not take on all responsibility for security. Users of all types still have a part to play.

Users and Evaluated Systems

General

6.4.15 As far as security is concerned, users can be considered to be of two types: trusted and untrusted.

6.4.16 Administrators would usually be regarded as highly trusted, because the special system privileges these users require in order to do their jobs, and the physical access they are allowed to the system, mean that the security of the system depends critically on them performing their duties responsibly.

6.4.17 End-users would usually be regarded as less trusted, and therefore will be provided with restricted access to system functions concerned with security, and have a more limited role in maintaining the system's security.

Trusted Users

6.4.18 The following are examples of security-related tasks which trusted users may engage in:

a) creating and deleting userIDs;

b) configuring the system;

c) choosing file access permissions;

d) checking audit logs to search for attempted security breaches.

6.4.19 An evaluated system should be provided with administration, delivery, configuration, start-up and operation documentation. During the evaluation the ITSEF evaluators will have checked that this documentation is accurate and will, if followed, maintain security. Therefore, trusted users should follow this documentation closely while performing their security tasks.

6.4.20 An evaluated system is always provided with a security target which defines the required environment in order to be secure, in conformance with the evaluation results. Trusted users are responsible for maintaining this operational environment to maintain the level of assurance confirmed by the evaluation results.

Untrusted Users

6.4.21 The following are examples of security-related tasks which untrusted users may engage in:

a) logging in to the system;

b) logging off from the system;

c) choosing passwords;

d) choosing access permissions to files they own.

6.4.22 These are not as vital to security as trusted users' tasks, but failure to perform them properly may imperil the security of the user's data, or even that for the system as a whole.

6.4.23 An evaluated system should be provided with user documentation. During the evaluation the ITSEF evaluators will have checked that this documentation is accurate and will, if followed, maintain security. Therefore, untrusted users should follow this documentation closely while performing their tasks.

Requirements Definition

6.4.24 During the initial requirements definition activity, a system accreditor may be asked to provide advice on the security policy to be applied, or may be involved in the development of the security target for an IT system.

6.4.25 It is usual for a system accreditor to be asked at this stage to approve the approach to be used in the development of the system; it may therefore be necessary to perform an extensive security assessment at an early stage in the project.

6.4.26 A high level security assessment of a system will typically be based on the techniques of risk analysis, relating all the possible threats to the **countermeasures** provided. A number of proprietary and government techniques exist which can be used to perform this analysis [BDSS], [CRAMM], [GISA2], MARION and MELISA. These techniques concentrate on functional security, and provide little or no guidance on the level of confidence required in the correctness and effectiveness of the countermeasures. However they do include aspects of non-IT security countermeasures, which are of interest to a system accreditor.

6.4.27 The system accreditor needs to ensure that the total set of countermeasures identified by risk analysis will be in place, and that they will work effectively together to satisfy the security policy. This analysis is analogous to the effectiveness assessment performed in an IT evaluation against the ITSEC, but includes non-IT countermeasures.

6.4.28 Some of the countermeasures will be provided by IT components of the system; their security features will either be defined in a security target for all IT aspects, or in a number of security targets for separate system components. In the latter case the system accreditor will have to ensure that the IT components will work together effectively in the system.

6.4.29 The system accreditor will also need to establish the level of assurance or confidence required in the security of the system, in addition to defining the security functionality required. Current techniques use a qualitative assessment of the level of risk to the system in order to assign a required confidence level.

6.4.30 Although these guidelines can be used in other fields, they have been developed for military-type applications, and are concerned primarily with just the confidentiality aspect of security. More work is required to extend this guidance to other aspects of security and other fields of application.

6.4.31 A particular problem for system accreditors is the determination of the required evaluation level of IT components of a system, in the case when several are included in the system design.

6.4.32 System accreditors may need to arrange training for users of secure systems.

6.4.33 System accreditors will need access to a wide range of information about the system, including typically:

a) system specifications;

b) system and higher-level security policies;

c) definitions of non-IT security functions;

d) security targets for IT components;

e) operational procedures documentation for the system, including those for IT components;

f) certificates/certification reports (and perhaps ETRs) for pre-evaluated components.

System Acceptance

6.4.34 While a system is being developed, or its IT components are being evaluated, changes may be proposed and vulnerabilities may be discovered. Problems found during evaluation are reported using the **problem reporting** mechanism defined in the national scheme.

6.4.35 System accreditors will be required to consider the security implications of reported problems and proposed changes. After the evaluation(s) has been completed, the system accreditor will be involved in the decision on whether a system can be operational. Both these cases require the system accreditor to provide a judgement on whether the required level of security has been, or will be, achieved.

6.4.36 In order to do this, it may be necessary to repeat elements of the analysis described in the previous section, but with more definite information on the actual or proposed implementation of the system.

6.4.37 The system accreditor will need to establish whether any **exploitable vulnerabilities** in the IT components (documented in the ETR) are adequately countered by non-IT measures that are in place, or whether additional non-IT measures need to be added before the system goes operational.

System Accreditation Maintenance

6.4.38 During the operational life of a system changes will be made to its configuration, components and operational use. These changes will need to be assessed by an accreditation authority to establish whether the security requirements are still satisfied.

6.4.39 Annex 6.D addresses the way in which the need for re-evaluation of evaluated IT components is established. An analogous procedure needs to be applied by system accreditors, but extended to include the non-IT aspects of the system.

Annex 6.A Evaluation Deliverables

Introduction

6.A.1 This annex summarises and explains the deliverable requirements of the ITSEC and is aimed, in particular, at sponsors and developers.

Responsibility for Deliverables

6.A.2 The responsibility to provide all the required deliverables for an evaluation lies with the sponsor. However, most of the deliverables will be produced and supplied by the developer (where the sponsor is not the developer). It is therefore advisable for the sponsor's contract with the developer to include details of what the developer is required to produce, and of what the consequences of failing to produce adequate deliverables will be.

6.A.3 For a particular arrangement between sponsor and ITSEF, the following details may have to be clarified:

a) the medium and format of computer-readable deliverables;

b) the schedule for the deliverables' production;

c) the number of copies of deliverables to be supplied;

d) the position regarding draft deliverables;

e) arrangements for any products to be used in conjunction with the TOE;

f) arrangements for discussing the development environment with the developer;

g) access to the operational and development sites;

h) type and duration of developer support, including computer access and requirements for office accommodation for evaluators.

6.A.4 In many cases the evaluators will require access to information provided by subcontractors, or third parties. The arrangements should take such cases into account.

6.A.5 The running costs and risks (e.g. loss or damage through fire, flood, theft, etc) in all deliverables should be the responsibility of the sponsor, unless specifically agreed otherwise with the evaluators. It should be noted that some deliverables, such as new or special purpose types of hardware may not have an easily identified replacement cost and may well present insurance risks that cannot be transferred to evaluators.

Management of Deliverables

Draft Deliverables

6.A.6 Stable and formally issued versions of deliverables are required for evaluation. There may be occasions, however, when it may be helpful for the evaluators to also see draft versions of particular deliverables, e.g.:

a) test documentation, to allow the evaluators to make an early assessment of tests and test procedures;

b) source code or hardware drawings, to allow the evaluators to assess the application of the developer's standards.

6.A.7 Draft deliverables are more likely to be supplied where the evaluation of a TOE is performed concurrently with its development. However, they may also be supplied during the consecutive evaluation of a product or system where the developer has had to perform additional work to address a problem identified by the evaluators (e.g. to correct an error in construction) or to provide evidence of security which is not provided in the existing documentation (e.g. effectiveness deliverables in the case of a product or system not originally developed with evaluation in mind).

6.A.8 It is recognised that developers are generally reluctant to supply draft deliverables to evaluators. However, it is in the sponsor's interest to provide drafts since the developer may receive early feedback of security-related deficiencies or faults, which may reduce the amount of any subsequent redevelopment work.

Configuration Control

6.A.9 For evaluation to E1, the sponsor need only supply a configuration list identifying the version of the TOE to be evaluated. Where the target evaluation level is E2 or above, the deliverables required by the evaluators must also:

a) be kept under configuration control;

b) be uniquely identified (e.g. by version number).

6.A.10 This requirement applies to all tangible deliverables, including, for example, all required evidence of effectiveness or correctness, such as a description of how a TOE's architectural design will provide the security enforcing functions of the security target.

6.A.11 Changes to deliverables should be kept to a minimum. Revised deliverables must be sent to the evaluators at the earliest opportunity.

The Security Target

6.A.12 It is the sponsor's responsibility to define what the security target will be. The objectives of a security target are as follows:

a) to provide a specification of a TOE's security functionality;

b) to relate a TOE to the environment in which it is intended to operate;

c) to provide the basis for the evaluation.

6.A.13 The intended audience for a security target is therefore:

a) the developer of the TOE: the security target defines the security requirements of the TOE;

b) the evaluators: the security target provides the baseline against which the TOE is evaluated;

c) the user of a TOE (i.e. those responsible for managing, purchasing, installing, configuring and operating the TOE): the security target gives all the information required to assess the suitability of the TOE for an intended application.

6.A.14 The required content and specification style of a TOE is determined by whether the TOE is a system or a product and the target evaluation level. It can be summarised as follows:

- System Security Policy Product Rationale;
- a specification of the required security enforcing functions;
- a definition of required security mechanisms (optional);
- the claimed rating of the minimum strength of mechanisms;
- target evaluation level.

Evaluation Deliverables

General

6.A.15 The general requirements for deliverables are given in figures 6.A.1 and 6.A.2. However, some additional deliverable requirements are implied, rather than explicitly described, in the ITSEC. In particular, the following deliverables, associated with the development environment in general, are usually required:

a) access to previous evaluation results (e.g. for re-evaluation of a TOE or where an evaluated product is a component of the TOE);

b) access to the development site, including access to the development tools, and facilities for interviewing (some of) the development staff;

c) access to the TOE in its operational environment;

d) technical and logistical support from the developer.

6.A.16 It is not necessary for each effectiveness deliverable to be a separate document. It is possible, and may in some cases be preferable, to have a single document which covers effectiveness as a whole.

Use of Products as Components of a TOE

6.A.17 One of a number of alternatives may be adopted for supplying deliverables related to a product which forms a security enforcing or security relevant component. Examples include:

a) the results of any previous evaluation of the product may be supplied;

b) the product may be treated in the same way as the rest of the TOE, in which case the appropriate deliverables relating to the product should be supplied.

6.A.18 The approach adopted for a particular evaluation must be acceptable to the certification body, sponsor and evaluators. If existing evaluation results are to be re-used, additional guidance is provided in part 4, chapter 4.6.

Development Environment

6.A.19 The evaluators will require documentation relating to the configuration control, programming languages and compilers, and developer's security used or applied during the development of a TOE. The evaluators will also require documentation relating more generally to the procedures, methods, tools and standards used during the development of the TOE, e.g.:

a) a quality plan, including development procedures;

b) details of the development methods used;

c) details of the development tools used;

d) software coding standards.

6.A.20 The evaluators will require evidence of adherence to procedures and standards, and evidence that methods and tools have been used correctly, e.g.:

a) configuration management plan;

b) configuration control records;

c) minutes of design reviews.

6.A.21 The evaluators may also require to make one or more specific visits to discuss the development environment with the developer. Topics to be discussed on such visits are listed in figure 6.A.3.

6.A.22 The evaluators have no right to access anything that is related solely to financial, contractual or staff issues (other than staff issues within the scope of the developer's security criteria of the ITSEC).

Operational Environment

6.A.23 The evaluators will require documentation relating to the use, administration, delivery, configuration, start-up and operation of the TOE.

6.A.24 The evaluators will require access to the operational TOE in order to perform penetration testing. Where the TOE is a system, the evaluators will also require access to the operational site, where possible, in order to:

a) discuss aspects of the operational procedures with representatives of the users;

b) perform penetration testing in the operational environment.

6.A.25 Where the TOE is a product, the evaluators will require access to a working implementation of that product in order to perform penetration testing. The sponsor can make the TOE available at the development site or, alternatively, the necessary equipment can be loaned to the evaluators and the penetration tests run within the ITSEF.

Evaluation Support

6.A.26 The evaluators may require logistical, consultancy and training support from the sponsor and developer during an evaluation.

6.A.27 A named individual in the developer's organisation should act as the point of contact for all developer support. This individual, or nominated alternatives:

a) should be able to provide support in a timely manner;

b) should be able to liaise with other development staff, as necessary, where detailed information is required on particular aspects of the TOE.

6.A.28 The total amount of support required will be evaluation dependent. Factors affecting the amount of support will include the target evaluation level, size and complexity of the system, and whether the developer and/or sponsor have previous experience in the development of evaluated systems and products. Some aspects of the evaluation process will demand more intensive support such as performing tests on the TOE.

6.A.29 The type of support required could include:

a) training;

b) informal discussions;

c) computer access and support;

d) office accommodation.

6.A.30 Informal training, preferably from someone in the development team, may be required in a number of proprietary areas where documentation is not widely available, such as:

a) the hardware and operating system(s) used in the TOE and its development;

b) development methods used;

c) development tools used.

6.A.31 The developer is not normally required to organise formal training courses specifically for the evaluators. However, the evaluators may wish to attend any training courses provided for other staff, e.g. where:

a) development staff are being trained, e.g. on a particular development method;

b) user training is being provided, e.g. on the security administration of the TOE.

6.A.32 Informal discussions with the developer may be required on any aspect of the TOE. Typically the evaluators may require the developer to provide a short description of a particular part of the TOE and then to answer any questions from the evaluators.

6.A.33 The evaluators will require access to suitable computer(s), principally in order to perform tests on the TOE. "Computer(s)" in this context includes any equipment used by the developer to build the TOE and to test the TOE.

6.A.34 Where the TOE is a system, the evaluators will also require, where possible, access to the computer(s) used in the TOE's operation (see Paragraphs 6.A.23 to 6.A.25).

6.A.35 The evaluators will require dedicated computer access for some of the time when performing additional tests (to the developer's) or penetration testing.

6.A.36 The duration of computer access will depend on the nature of a particular TOE.

6.A.37 Access to a computer will normally be at the development or operational site. In some cases, however, it may be convenient to provide access in an alternative location, for example by the supply of a computer to an ITSEF.

6.A.38 When the evaluators are using a computer, support may be required to provide help with basic operations, such as starting up the computer, making backup copies of the TOE, running tests, etc.

6.A.39 Office accommodation for the evaluators' sole use should be obtained as required for when the evaluators are working at the development or operational site. This accommodation should be capable of supporting the required number of people and should include:

a) basic furniture, including a telephone;

b) secure storage facilities for information of a classification level appropriate for the TOE.

6.A.40 It is recognised that site rules may prohibit general unescorted access on the development or operational site. However, the evaluators will require privacy at times when working at a site, and so arrangements to allow the evaluators to be unescorted when in this office will need to be made.

Figure 6.A.1 Evaluation Deliverables (Effectiveness)	
DELIVERABLE	**ALL EVALUATION LEVELS**
Suitability analysis: **an investigation showing that the security enforcing functions and mechanisms of the TOE will in fact counter the threats to the security of the TOE identified in the security target**	###
Binding analysis: **an investigation showing that the security enforcing functions and mechanisms of the TOE bind together in a way that is mutually supportive and that provides an integrated and effective whole**	###
Strength of mechanisms analysis: **an investigation showing the ability of the TOE as a whole to withstand direct attacks based on deficiencies in its underlying algorithms, principles or properties; this assessment will require consideration of the level of resources that would be needed for an attacker to execute a successful attack**	###
List of known vulnerabilities in construction: **a list of potential vulnerabilities in the construction of the TOE (identified by the developer) plus an argument for why they are not exploitable**	✓
Ease of use analysis: **an investigation showing that the TOE cannot be configured or used in a manner which is insecure but which an administrator or end-user of the TOE would reasonably believe to be secure**	###
List of known vulnerabilities in operational use: **a list of potential vulnerabilities in the operation of the TOE (identified by the developer) plus an argument for why they are not exploitable**	✓

Figure 6.A.2 Evaluation Deliverables (Correctness)

Deliverable	Evaluation Level					
	E1	E2	E3	E4	E5	E6
Requirements						
The security target for the TOE	###	###	###	###	###	###
Definition of or reference to an underlying formally specified model of security				###	###	###
Informal interpretation of the underlying model in terms of the security target				###	###	###
Architecture						
Informal description of the architecture of the TOE	###	###	###			
Semiformal description of the architecture of the TOE				###	###	
Formal description of the architecture of the TOE						###
Detailed Design						
Informal description of the detailed design		###	###			
semiformal description of the detailed design				###	✓	✓
Implementation						
Test documentation	(###)	###	###	✓	###	###
Library of test programs and tools used for testing the TOE		###	###	✓	###	
Library of test programs and tools used for testing the TOE, including tools which can be used to detect inconsistencies between source code and executable code if there are any security enforcing or security relevant source code components (e.g. a disassembler and/or a debugger)	(###)					✓
Source code or hardware drawings for all security enforcing and security relevant components			###	###	###	✓
Informal description of correspondence between source code or hardware drawings and the detailed design			###	###	□	
Informal description of correspondence between source code or hardware drawings and the detailed design and the formal specification of security enforcing functions						✓

Note: (✓) - optional deliverables

Figure 6.A.2 Evaluation Deliverables (Correctness)

Deliverable	Evaluation Level					
	E1	E2	E3	E4	E5	E6
Configuration Control						
Configuration list identifying the version of the TOE for evaluation	✓	✓	###	###	###	▯
Information on the configuration control system		✓	###			
Information on the configuration control system and its tools				###	###	###
Audit information on modification of all parts of the TOE subject to configuration control				###		
Audit information on modification of all objects of the TOE subject to configuration control					###	###
Information on the acceptance procedure			###	###	✓	###
Information on the integration procedure					✓	###
Programming Languages and Compilers						
Description of all implementation languages used			✓	###	###	###
Description of all compilers used				###	###	###
Source code of all runtime libraries used					###	###
Developers Security						
Information on the security of the development environment		###	###	###	###	###
Operation						
User documentation	###	###	###	###	###	###
Administration documentation	###	###	###	###	###	###
Delivery and configuration documentation	###	###	###	###	###	###
Startup and operation documentation	###	###	###	###	###	###

Figure 6.A.3 Development Environment Discussion Topics

DEVELOPMENT CONFIGURATION CONTROL

Scope: The procedures (manual and automated) for the control and traceability of project material

Topics:
- Computer organisation
  - Directory structures
  - Software library and access control
- Change control
- Release procedures

PROGRAMMING LANGUAGES AND COMPILERS

Scope: The programming languages used for implementation

Topics:
- Definition of languages
- Implementation dependent options
- Compilers

DEVELOPMENT SECURITY

Scope: Security of the development environment, i.e. protection of the TOE and confidentiality of associated documents

Topics:
- Physical measures
- Procedural measures
- Personnel measures

DEVELOPMENT METHODS

Scope: The different phases of the development and the approach adopted

Topics:
- Project history and current status
- Representations produced
- Design process
- Coding phase
- Test strategy

DEVELOPMENT TOOLS

Scope: The tools (proprietary and purpose-built) used during development

Topics:
- Development computers, system management
- Compilers/linkers/debuggers
- System generation procedures
- Test harnesses

DEVELOPMENT PROCEDURES

Scope: The controls applied during development

Topics:
- Project management procedures
- Quality assurance procedures
- Technical assurance procedures

DEVELOPMENT STANDARDS

Scope: The standards used during the development

Topics:
- Design standards
- Coding standards
- Documentation standards

Annex 6.B Writing a Security Target

Introduction

6.B.1 This annex presents guidance for an evaluation sponsor for writing a security target. By way of illustration, a description is presented of how the SWAN system security target (shown in part 5 of the ITSEM) may be re-written.

The Purpose of a Security Target

6.B.2 The sponsor's objective in creating a security target is to provide a complete and consistent baseline for use in the evaluation. The security target is a comprehensive document (or set of documents) which states, among other things:

a) the security objectives of the product or system (TOE);

b) the countermeasures employed by the TOE to address the perceived threats.

6.B.3 To this end, a security target presents the security requirements for the TOE, described at a high level abstraction.

6.B.4 In addition, the security target forms part of the contractual baseline between the sponsor and the ITSEF, specifying information, such as the evaluation level, which is of relevance throughout the evaluation.

6.B.5 From the developer's point of view, the security target is an integral part of the high level specification of the TOE. To this end, the developers require the security target to state unambiguously capabilities and possible uses of the TOE.

6.B.6 The security target may contain procedural, functional and technical aspects. It may also address other aspects imposed by the requirements, such as support.

6.B.7 The security target constitutes a specification for the security enforcing parts of the implementation. The security target should be written as early as possible in the development life cycle, to allow a concurrent evaluation to start early in the development. However, this is only possible if the security target is sufficiently stable.

6.B.8 Although the security requirements have to be separately specified in the security target, the refinement of security and non-security requirements is performed concurrently.

6.B.9 For a consecutive evaluation, if the TOE has been developed before the definition of the security target, it will be necessary to "reverse engineer" all of the required information.

6.B.10 Although the sponsor is responsible for providing the security target to evaluators, he may not be expert in all aspects of security. It is therefore recommended that the sponsor be assisted when writing a security target. Such assistance can be provided by developers, who are well placed to produce the specification they have to implement. However, an ITSEF could also be consulted to provide guidance regarding the security target content and presentation.

6.B.11 For procurers of a system (i.e. those users whose requirements the system is designed to satisfy), the security target should provide a basis for their procurement decisions.

6.B.12 It is the responsibility of the evaluators to determine whether the TOE is consistent with the security target. In addition, the evaluators must assess whether the TOE specification is valid in the context of the other evaluation deliverables.

The Content of a Security Target

6.B.13 To satisfy its role in an evaluation, a security target must:

a) detail the security requirements of the TOE;

b) state the countermeasures proposed to address the perceived threats to the assets protected by the TOE.

6.B.14 For this purpose, the ITSEC mandates that:

a). security requirements are covered in a security policy (for systems) or a product rationale (for products);

b) functions are devised to address the security requirements; these are called security enforcing functions;

c) should a particular technical approach to meeting a security requirement be mandated, for instance the use of a specific password encryption algorithm, it will be listed as a required security mechanism;

d) a claimed rating for the minimum strength of mechanisms is provided, which shall be one of *basic*, *medium* or *high*;

e) a target evaluation level for the TOE is specified.

6.B.15 It should be noted that, depending on the choice of the evaluation level, the security enforcing functions may have to be specified in a semiformal or formal style.

Risk Analysis

6.B.16 The specification of security enforcing functions involves a compromise between the need to protect assets and the cost of such protection (e.g. financial, human resources, operational, technical). This compromise is controlled by a risk analysis process.

6.B.17 In addition, the construction of a TOE addresses both the requirements of the TOE and any relevant constraints on the TOE (e.g. laws, instructions, technology, assets, cost, etc).

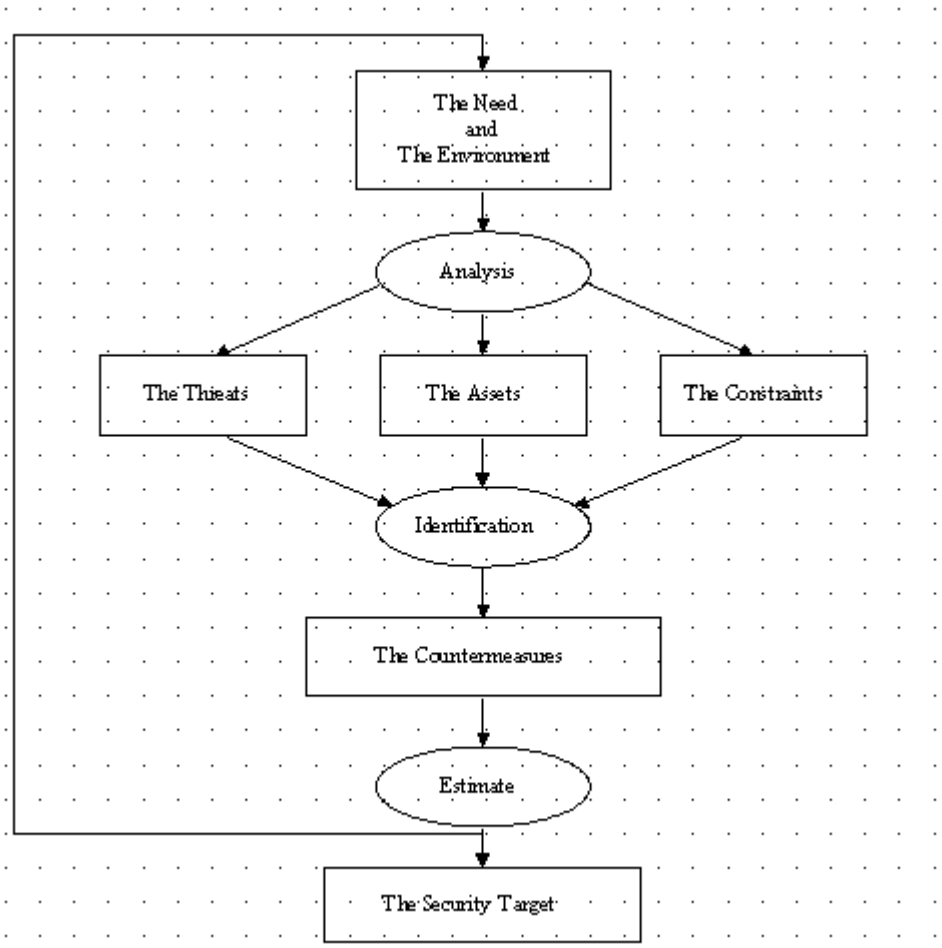


Figure 6.B.1 The Risk Analysis Approach

6.B.18 Risk analysis determines the threats to the assets protected by the TOE. For each threat the probability that it may result in an asset being compromised is assessed.

6.B.19 The risk analysis should be performed as one of the first activities in the TOE development.

6.B.20 However, development is seldom a linear process, so periodic revisions of and changes to the security target are likely. Such changes are a problem for evaluators, often invalidating evaluation results.

6.B.21 The risk analysis process guides the production of the security target, addressing assets, threats and countermeasures in order ultimately to produce the security target.

6.B.22 Risk analysis consists of a series of activities which are performed on the specifications and the requirements (see figure 6.B.1). These activities are:

a) problem analysis (concerned with environment and needs);

b) option identification (concerned with assets, threats and constraints);

c) solution estimation (concerned with convenience, feasibility and costs of countermeasures);

d) decision integration (concerned with choices and reporting).

6.B.23 Variants of this process are described in standard methodologies ([CRAMM], MARION, MELISA, [GISA2]). They are most useful in producing directories of resources, threats and countermeasure classes.

6.B.24 In the absence of a methodology, it may be appropriate to use generic specifications. The pre-defined functionality classes in the ITSEC and the ISO open systems security models are examples of this.

System Security Policy or Product Rationale

General

6.B.25 The security target starts with a statement of the threats, objectives and environment of the TOE. For a system, this is done in a system security policy. For a product, this is done in a product rationale.

6.B.26 The security policy or product rationale states who may do what with the facilities, services, functions, and devices of the TOE.

6.B.27 Production of a security policy or a product rationale for use in a security target can be difficult. A security policy or product rationale should express, without considering the design of the TOE, the assets to be protected and the rules governing the handling of the assets.

Intended Environment

6.B.28 A study of the TOE and the environment in which it will operate, including a risk analysis of the security aspects of the TOE, defines the operational characteristics of the TOE. These characteristics determine how the TOE interfaces with its environment and therefore should be described in the security target.

6.B.29 This section of the security target should define the:

a) purpose and boundary of the TOE;

b) information to be handled by the TOE, and how it is to be handled;

c) personnel using the TOE (e.g. users, operators, administrators, etc);

d) the equipment necessary to support the TOE's operation;

e) location and topology of the TOE, including physical security measures;

f) operational modes and procedures;

g) organisation and its procedures.

The SWAN System: Intended Environment

6.B.30 The Site-Wide Area Network (SWAN) is a communications network which permits various user communities to access different data processing applications.

6.B.31 The example system is situated in a large site belonging to a commercial organisation. The site is completely enclosed by a perimeter fence which is well guarded. All personnel have been vetted by the organisation and are considered trustworthy. Visitors to the site are escorted at all times.

6.B.32 Within the site, there are a number of areas which offer further protection in the form of physical access control and other procedural security mechanisms. There is a low TEMPEST and cryptographic threat. Terminals are located in secure rooms and personnel will be prevented, by the authorised users, from using a terminal unsupervised in a room that they are visiting.

6.B.33 The site contains a variety of different IT systems, procured at different times, from different manufacturers and used for a variety of purposes, such as transaction processing, billing and company administration.

6.B.34 Connections between user terminals and host computers, which may be located in different buildings, were once made via dedicated fibre-optic cables. They have now been replaced by the SWAN. The SWAN is a TCP/IP token ring network consisting of a dual counter rotating backbone and various sub-rings. End system equipment is attached to the SWAN by host access points or terminal access points.

6.B.35 Host machines are operated in either dedicated or system-high mode, e.g. at Company Confidential, Management in Confidence or Directors in Confidence.

6.B.36 Access rights are defined for each user. All personnel on the site are either authorised to access at least Company Confidential information, or are escorted by authorised personnel.

6.B.37 Operating procedures are derived from a previous configuration when each server was the hub of a dedicated network. As a consequence, access to applications supported by each host is locally managed on a discretionary basis by an Application Manager.

6.B.38 While every terminal and host may operate at different security levels, the system authority in the new version of SWAN has established a mandatory access control policy for linking terminals to servers.

Security Objectives

6.B.39 The first step in establishing a security policy is to determine the security objectives. The security objectives are expressed in terms of:

a) The organisation's assets requiring protection, be it by the TOE, some other system or perhaps by manual/physical means; the assets include the information to be handled by the TOE, processes to be automated by the TOE and the responsibilities and/or roles of the users.

b) The TOE's resources, as defined in the external specification; the resources may be the physical resources, for instance equipment or devices, or abstract resources, such as the TOE configuration, processes, algorithms or code.

6.B.40 Risk analysis considers the level of security provided by the security objectives (for instance, in the case of data confidentiality, what classification can be protected). Evaluation does not consider this, but instead concentrates on the assurance that can be gained in the implementation of the security enforcing functions. Therefore, the security target should make no reference to this degree of protection.

6.B.41 Two approaches are possible when considering security objectives:

a) all data and resources are analysed in sequence, considering all relevant security objectives;

b) resources and data related to the same security objective are grouped together for analysis.

6.B.42 Availability objectives are described in terms of status, capabilities, service duration, response times, priorities, and degradation tolerance.

6.B.43 Integrity objectives are described as one of:

a) conformance to standards, specifications and references;

b) conformance to an initial state or condition;

c) rules to be observed for consistency and coherence.

6.B.44 Confidentiality objectives explain the expected use of each resource, rather than addressing vulnerabilities to be avoided (e.g. disclosure, context substitution, goal tampering).

6.B.45 The author of the security target should endeavour to make this section as complete as possible, since the security objectives ultimately form the baseline for the evaluation. Any feature of the TOE which cannot be traced back to a security objective cannot be considered security enforcing.

The SWAN System: Security Objectives

6.B.46 The assets of the organisation to be protected are application services delivered to each user community. There is no security objective for the system resources.

6.B.47 These services (information and processing) should not be available to persons outside the community.

6.B.48 There is no availability objective.

6.B.49 There is no integrity objective, which implies that an attack against integrity of services procured for each community is not expected, or that such an attack is tolerable as long as confidentiality of the services is not endangered.

6.B.50 Improper use of application services by authorised users is not of concern.

The Threats

6.B.51 The next step in establishing a security policy is to determine the perceived threats to the assets, i.e. actions that may violate the security objectives.

6.B.52 As was the case for security objectives, threats are an issue to be considered during the specification of the TOE. As suggested above, they relate to the external description of the TOE. However, threat assessment is more difficult than determining the security objectives since it is impossible to address all possible modes of attack.

6.B.53 Risk analysis methods can be helpful during threat assessment, not so much for the systematic process employed, but for the knowledge that can be gained from them. The techniques can provide a list of generic threats that can be readily applied to the TOE concerned. This can provide suitable guidance for a threat assessment, which can be event-directed or goal-directed, according to the analyst's requirements.

6.B.54 Security target authors should note that they are responsible for the accuracy and completeness of the security objectives and threats. Evaluators cannot verify the completeness of this information but will verify the accuracy and the consistency.

The SWAN System: The Threats

6.B.55 During risk analysis, different kinds of threats have been considered in turn:

a) physical attacks against the system and its environment;

b) interception of emanated radiation;

c) direct attack against applications.

6.B.56 Physical attacks do not apply to hosts or terminals that are permanently guarded or monitored, but do apply to network cables liable to unauthorised connections.

6.B.57 Emanated radiation was not a threat due to TEMPEST screening of the buildings and the use of fibre-optic cables.

6.B.58 An attack may therefore be perpetrated only locally through the network:

a) a user may try to access a service to which he is not authorised;

b) a user may masquerade as another user.

6.B.59 Authorised users are considered trustworthy, hence there is no risk of misuse of the system or collusion with an attacker.

System Security Policy

6.B.60 For a system evaluation, the actual operational environment is known and the threats to the system can be predicted. Existing countermeasures (which may be some combination of electronic, physical, procedural, and personnel countermeasures) can be taken into account and the security objectives of the system can be derived by the sponsor. This information is provided by a system security policy.

6.B.61 An organisation typically has several security policies. There usually exists a security policy at each level of an organisation according to the assets pertinent to that level. For example, an organisation's IT system will normally have a security policy which will specify the rules appropriate to preserve information handled by the system and the system's components (eg. data, devices, processes, etc.).

6.B.62 The security policies should refine the amount of detail at each level of the organisation. For instance, the protection of sensitive information need not be specified in the initial organisational security policy but instead considered gradually and iteratively in the lower level security policies and their security enforcing functions definition (see figure 6.B.2).

6.B.63 The system security policy defines the laws, rules and practices that control how sensitive information and other resources are managed within the system. Unlike the technical security policy, it includes physical, personnel and procedural measures.

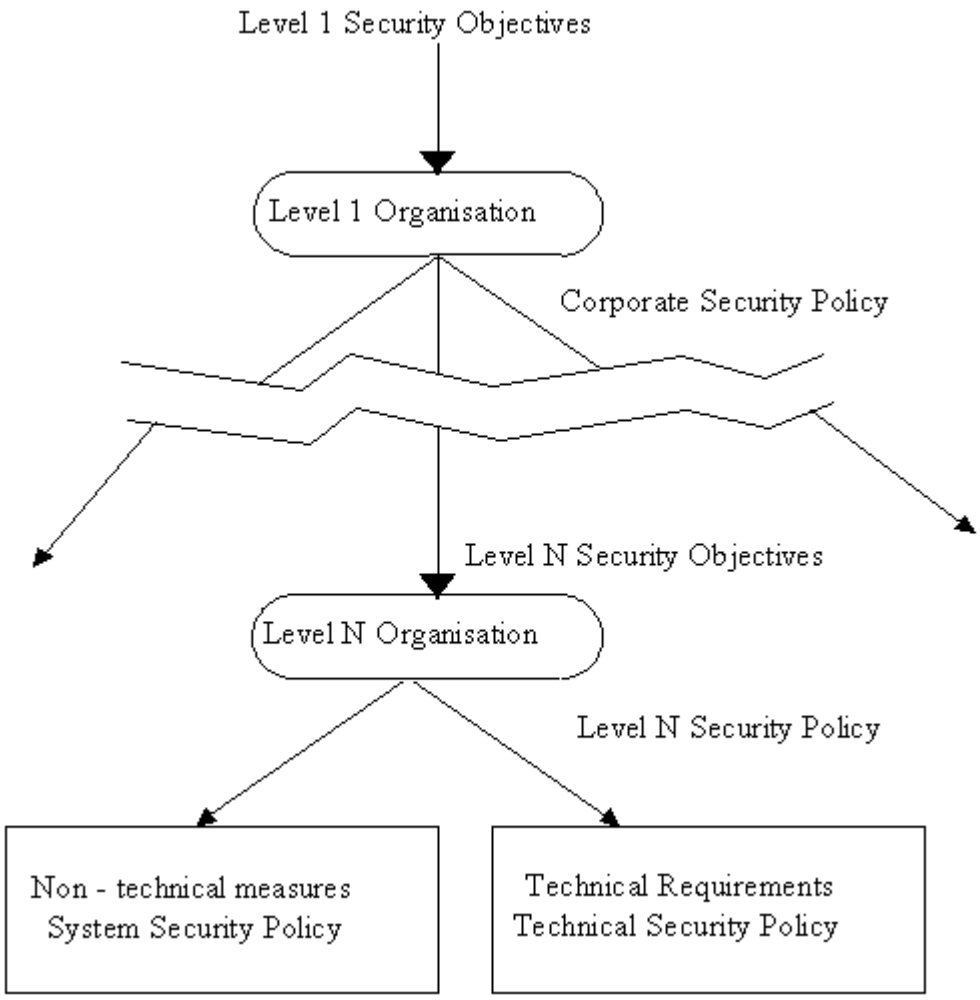


Figure 6.B.2 Derivation of a Security Policy

6.B.64 The technical security policy defines rules controlling the processing of sensitive information and the use of resources within the system itself.

6.B.65 The security policy proper establishes a link between the security requirements established in the `threats' and `objectives' steps, and the security enforcing functions defined later in the security target. From the organisation's point of view, the information already contained in the security policy is sufficient to form an implementation specification. However, the information requires further refinement before it can constitute a requirements specification for the TOE. This refinement is the objective of the last step in establishing a security policy.

6.B.66 The security objectives and perceived threats suggest rules controlling the various users of the TOE.

6.B.67 The rules state which:

a) operations are mandatory, allowed or forbidden for each asset;

b) roles may, must or must not undertake these operations.

6.B.68 The rules represent the organisation's response to the requirement for security and result from:

a) general security practices;

b) doctrines within the organisation;

c) plans specifically designed to cope with the problem concerned.

6.B.69 Furthermore, the following general security principles must be adhered to:

a) separation of roles/users, which aims to limit the possibility of attack resulting from the propagation of privileges from one user to another; this is particularly pertinent to the removal of roles/users from a system;

b) ease of use, which aims to avoid mistakes in the operation of the TOE which may give rise to vulnerabilities;

c) protection by default, which aims to avoid the requirement for active measures to maintain security;

d) exception elimination, which aims to make the model of security easy to understand and adhere to;

e) least privilege, which aims to minimise the risk of abuse by requiring that the level of authorisation assigned (to a user, role ,process, etc) is just sufficient to perform its task.

6.B.70 The completed security policy must be internally consistent and address all security objectives and threats.

6.B.71 The ITSEC requires the security policy rules to be segregated into two subsets:

a) non-technical measures that consist of physical, personal or procedural measures established to control the environment in which the TOE operates (e.g. the system security policy);

b) technical measures which constitute the security requirements from which security enforcing functions may be developed (e.g. the technical security policy).

The SWAN System: System Security Policy

6.B.72 A first high level statement of the system security policy could be summarised by the following rules:

a) a user may have access to authorised services;

b) a person must not have access to unauthorised services.

6.B.73 It should be noted that:

a) this statement, through the `authorisation' concept, implies an Administrator for assignment and verification of authorisation;

b) authorisation, in turn, introduces an extra set of resources which are integrity sensitive;

c) the statement is not rigorous since it does not assign a duty to the user.

6.B.74 Rule (a) only concerns users and does not consider the system, which according to the initial hypothesis (no availability objectives), is not required to provide any particular level of service. Thus, rule (a) is obviously a non-technical measure.

6.B.75 Rule (b), on the other hand, refers to the system, which is supposed to make this prohibition effective. It is therefore a technical security measure that should be re-written:

a) the system must deny access to unauthorised services.

6.B.76 This statement of security policy is at too high a level, since it does not yet take account of the system's external requirements. These requirements indicate among other things that this system is a network which opens links between terminals and hosts, as allowed by a mandatory policy, and that each application service is managed locally, on a discretionary basis by an administrator. All these are sensitive resources and duties which must be made explicit in a more detailed statement of the security policy. Rule (b) can be re-stated:

a) (1) the system must deny access to unauthorised links;

b) (2) the system must deny access to unauthorised hosts.

6.B.77 By making the threats explicit, rule (1) may be sub-divided into:

a) (1.1) the system must prevent intrusion into links;

b) (1.2) the system must deny access to unauthorised links according to a mandatory access control policy.

6.B.78 There are no rules corresponding to (1) and (2) about terminals, which are both dumb and monitored and are not considered under threat.

6.B.79 Rules defining the Administrator's duties may now be made explicit in two further rules:

a) (3) an Application Administrator may modify authorisations for services;

b) (4) the system must deny modification of authorisation by others.

6.B.80 The Network administrator's duties (including responsibility for security levels assigned to hosts) could be rendered explicit in the same way.

Formal Model of the Security Policy

6.B.81 The security policy must be independently verified by evaluators for consistency. The rules can be amended into mathematical form to facilitate verification. This leads to the concept of a Formal Model of the Security Policy, as required for evaluation at E4 and above.

Product Rationale

6.B.82 A product may be used in any number of different systems and operational environments and so the actual operational environment of a product is not known. The security target can only define an intended method of use and make assumptions about the operational environment in which it is to be used and the threats that its security enforcing functions are designed to encounter.

6.B.83 For a product, the security target will comprise a list of claims made about the TOE by the sponsor (usually the vendor of the product) aimed at providing a potential purchaser with sufficient information to determine whether a product is suitable to satisfy some or all of his system security objectives. This information is provided in the form of a product rationale.

6.B.84 A product may be designed to operate in a number of configurations. For example, a database package may operate on a stand alone system or may operate as a distributed database in a networked environment. For such products, it may not be desirable, or feasible, to evaluate all configurations, in which case the evaluators and sponsor must agree the configuration(s) to be evaluated. This must be documented in the security target.

6.B.85 The vendor of the product, after having performed some market research, must be able to claim that the product will be able to protect a specific asset (or set of assets) which exists in an intended environment. In addition, the vendor must be able to identify some threats (relevant in the intended environment) which the product is capable of countering.

Security Enforcing Functions

6.B.86 Security Enforcing Functions (SEFs) are, at the highest level of abstraction, a statement of the functionality required to satisfy the security objectives. The SEFs must provide a unalterable and non-circumventable whole which completely fulfils the requirements formulated in the security policy.

6.B.87 The first step in specifying the SEFs is to formulate a SEF for each individual rule in the security policy, establishing a one-to-one relationship between SEFs and rules. Such SEFs are termed *Operational SEFs* since they directly implement the security policy. Further SEFs can be formulated which provide functions to assist the operational SEFs. Such SEFs are termed *Support SEFs*.

6.B.88 Operational SEFs may be categorised as one of four types of function as follows:

a) Preclusion functions, which aim to prevent potential attacks by minimising assets; for instance, a system may be flushed of sensitive data between user sessions.

b) Detection functions, which aim to detect and keep traces of attacks.

c) Confinement functions, which aim to control access to sensitive resources; such functions may provide compartments, enforce protection masks or prevent access to transient data. Cryptographic mechanisms are often confinement functions.

d) Restoration functions, which provide support for the safe recovery of the TOE after failure or attack.

6.B.89 Once all operational SEFs have been formulated, the author of the security target could determine any support SEFs necessary. Such SEFs must ensure the operational SEFs operate correctly at all times and cannot be circumvented. Support SEFs are important since they provide protection for a subset of sensitive resources, i.e. the SEFs themselves. The determination of support SEFs is an iterative process which terminates when all SEFs (including the support SEFs themselves) are protected.

6.B.90 This iterative process is convenient for the development of the security target. However, the ITSEC makes no distinction between operational and support SEFs, but rather suggests that all SEFs should be classified under the following generic headings:

a) identification and authentication;

b) access control;

c) accountability;

d) audit;

e) object re-use;

f) accuracy;

g) reliability of service;

h) data exchange.

6.B.91 Such classification of SEFs is designed to facilitate the comparison between different TOEs.

6.B.92 The pre-defined functionality classes are a part of the ITSEC, so its generic headings tend to be used in preference to others.

6.B.93 Often a SEF will be relevant to more than one heading. In this case there will be a cross reference to other headings. If a particular generic heading is not relevant to the functionality class then it is omitted.

6.B.94 At this stage, the SEFs must be described at a level of detail adequate to show their correspondence to the underlying security policy.

The SWAN system: Security Enforcing Functions

6.B.95 To satisfy the Technical Security Policy statement the SWAN developers have proposed the following functions:

a) Links between terminals and hosts will be encrypted by `approved' devices located in front of the network's access points. Encryption keys are specific to end systems. This choice is stated to be effective against intrusion into the network and is a solution to rule (1.1).

b) A function controlling access to host machines is installed in the network. This function implements a mandatory access control policy whose effect is to prevent the opening of virtual circuits between a terminal and a host which are not at the same security level. This choice is a solution to rule (1.2).

c) Discretionary access control functions, as defined in the old dedicated solution for each server, are retained. This decision, dictated by external requirements, is representative of constraints which are specific to the system and environment; it is a solution to rule (2).

6.B.96 The developer proposes a fourth security function (cf. SWAN example in part 5 of the ITSEM), to implement authentication of users requesting access to the network. Is that function superfluous, considering that the security policy appears covered with only three functions? It is evident that the network access control (function 2) implies an authentication of the terminals. Access control is not achievable without an associated authentication. In which case, must access control be explicit?

6.B.97 If in general the response *is no*, the explicit statement may be deferred until the refinement of the access control. In the present case, the developer has chosen to use the equivalent user authentication rather than terminal authentication.

6.B.98 A complete presentation of the SWAN SEFs would require the examination of the support functions necessary to guarantee the correct operation of the four described operational functions. This presentation would include the measures adopted to verify these functions, to preserve the secret elements, or to avoid the bypassing of the network controls. These measures are justified by the protection of the newly introduced sensitive resources in the system definition. All of these problems are presented in part 5 of the ITSEM as implementation problems, but should also be considered in the security target.

quired Security Mechanisms

6.B.99 A security target may optionally prescribe or claim the use of particular security mechanisms, i.e. the devices, algorithms or procedures which should be used to implement certain SEFs. Such mechanisms are likely to include:

a) algorithms such as data encryption algorithms, hashing algorithms, error correction codes and password generation algorithms;

b) I&A mechanisms, such as biometrics (voice recognition, fingerprints) and PID devices.

6.B.100 Such mechanisms may be mandated by analyses performed during the specification of security requirements.

6.B.101 In general, the author of the security target should avoid over-specification of security mechanisms, which would mandate both security objectives and the measures adopted to achieve them.

6.B.102 Until now, the security target has specified the SEFs in an abstract manner with no reference to the implementing mechanisms. In practice, each SEF is realised by one or more mechanisms, each of which may address several SEFs.

6.B.103 When specifying required mechanisms, the author must consider whether the mechanisms are security relevant and hence whether they should have any place in the security target.

6.B.104 In principle, the specification of security mechanisms should be limited to cover only security requirements. These requirements may suggest the use of a particular technique, algorithm, component or development method They can even mandate the use of a particular product or developer.

6.B.105 Attributes which are not specified in the security target are implicitly considered as part of the implementation process and left to the developer. Such choices will have to be justified in the deliverables produced to support the evaluation.

The SWAN system: Required Security Mechanisms

6.B.106 The SWAN example does not mention required security mechanisms for the system installation. Nevertheless, one can imagine that the developer may wish to re-use the password authentication mechanisms which are built into the server.

Claimed Rating of the Minimum Strength of Mechanisms

6.B.107 A mechanism is the logic or algorithm which implements a particular security enforcing or security relevant function.

6.B.108 Some mechanisms have an underlying weakness in that they can be overcome by the use of resources, special equipment, or opportunity by an attacker. An example is an authentication system which can be defeated by successively guessing all possible passwords.

6.B.109 Such mechanisms can be rated as basic, medium or high, depending on the level of attack they can withstand (see annex 6.C for more information).

6.B.110 The security target should claim a rating for the weakest critical mechanism in the TOE.

The SWAN System: Claimed Rating of the Minimum Strength of Mechanisms

6.B.111 The required minimum strength of mechanisms of the SWAN system as a whole is *medium*.

6.B.112 To meet this requirement for the system, the developer has claimed the following individual strengths of mechanisms:

a) for the mechanism implementing countermeasure 1 (CM1), which access control, the claimed strength of mechanism is *high*;

c) for the mechanism implementing countermeasure 3 (CM3), which encrypts data on the links between terminals and hosts, a mechanism of *medium* strength, approved by the National Authority, is used;

d) for the mechanism implementing countermeasure 4 (CM4), which authenticates users logging on to a host, the claimed strength is also *basic*.

6.B.113 To justify these choices, the developer states in the security target that only the encryption mechanism is critical. This reasoning is correct since even if the access control mechanisms fail, the attacker merely gains access to encrypted data and is unable to violate the security objectives.

6.B.114 An analysis (not presented in part 5 of the ITSEM) was conducted and the following results were obtained:

a) the encryption mechanism was assessed by the National Authority and was found to be of *medium* strength;

b) the automatically generated passwords are 8 characters long with a validity of up to 60 days, with no measure to limit invalid password attempts; the network and application services authentication was rated *basic*;

c) management of the network or collusion with the Network Administrator are not considered here; without supporting authentication, the strict network access control was judged *high* in the absence of threats other than network monitoring.

The Evaluation Level

Choosing an Evaluation Level

6.B.115 The security target must specify a target evaluation level for evaluation of the TOE. This shall be one of the ratings E1, E2, E3, E4, E5 or E6.

6.B.116 The choice of evaluation level is a compromise between what is desirable (i.e. the greatest assurance), and what is possible, taking into account the costs. Not only do the financial costs of evaluation have to be carefully considered but also other costs such as producing and distributing the necessary deliverables.

6.B.117 Figures 6.B.7 and 6.B.8 summarise the effect of evaluation level on the content of security targets.

Information Required

6.B.118 Evaluation levels are distinguished by the granularity of the design information required for the evaluation. This is shown in the following table:

Figure 6.B.3 Level and Information

Evaluation Level	Information Required
E1	architectural design
E2	architectural design and detailed design
E3 and above	architectural design, detailed design, source code and hardware drawings

Specification Style

6.B.119 The different evaluation levels require different levels of specification, according to the following table:

Figure 6.B.4 Level and Style

Evaluation Level	Style of Specification
E1, E2, E3	informal documentation
E4, E5	underlying formal model of security policy, semiformal specification of the security enforcing functions and semiformal descriptions of the architecture and detailed design
E6	underlying formal model of security policy, formal specification of the security enforcing functions, formal description of the architecture and semiformal description of the detailed design

Rigour of Specification

6.B.120 The rigour with which the content, presentation and evidence are given is also dependant on the target evaluation level in terms of the *state*, *describe*, *explain* transitions. The requirements for each evaluation level are summarised in figure 6.B.5.

Figure 6.B.5 Rigour of Specification

Requirements for Evidence	Target Evaluation Level					
	E1	E2	E3	E4	E5	E6
	state		*describe*		*explain*	
Relevant facts provided	✓	✓	✓	✓	✓	✓
Characteristics enumerated			✓	✓	✓	✓
Justification given					✓	✓

6.B.121 For example, at E1 and E2, a security target may describe a logon process as follows:

> The <TOE> shall identify and authenticate authorised users by checking the validity of a PID, user ID and password. Consistency between the PID, user ID and password shall be checked. Users shall be allowed a maximum of three attempts to complete the log-on process successfully. If the number of attempts exceeds three, the failure shall be logged and the user shall be locked out of the system.

6.B.122 At E3 and E4, the process should be described in more detail by enumerating the characteristics of the logon process. The security target might include statements as follows:

> The <TOE> shall identify and authenticate authorised users by checking the validity of a PID, user ID and password. The system shall check that:
>
> a) the user ID entered at the keyboard matches that held in machine readable form on the PID;
>
> b) the user ID is recorded in the User Authorisation File;
>
> c) the password is valid for the user-ID.
>
> If the number of attempts exceeds three the system shall:
>
> a) write an audit trail message identifying the incident type (i.e logon failure), date and time of the incident, terminal identifier and user name;
>
> b) lock the user out of the system by disabling the user's entry in the User Authorisation File.

6.B.123 At E5 and E6, explanations are required which provide a justification for the specified functionality. The security target might include statements such as the following:

a) auditing failed login attempts warns the Security Officer that a particular terminal, a specific user account or the system as a whole, is under attack;

b) the user's entry in the User Authorisation File is disabled to prevent access to the system until authorised by the Security Officer.

Use of Tools

6.B.124 The different evaluation levels require different use of tools, as described in the following table:

Figure 6.B.6 Level and Tools

Ealuation Level	Tools Required
E1	none
E2 and above	test tools
E3 and above	well-defined programming languages
E4 and above	developers tools, tool based configuration control system
E6 and above	object code analysis tools

The SWAN System: Evaluation Level

6.B.125 A confidence level of E3 was decided upon. This provided an adequate amount of assurance and was achievable given the financial and time constraints.

Figure 6.B.7 Security Target for a Product Evaluation

1. Introduction

    1.1 Claimed Rating of Minimum SOM {2.25}

    1.2 Target Evaluation Level {2.2g6}

2. Product Rationale {2.16-2.17}

    2.1 Security Objectives

    2.2 Intended Method of use {2.17}

    2.3 Intended Environment {2.17}

    2.4 Assumed Threats {2.17}

3. Model of Security Policy {2.81-2.83}

4. Specification of Security Enforcing Functions {2.18-2.24}

    [Defn of Security Enforcing Functions]

    [Defn of Required Security Mechanisms (optional)]

    4.1 Identification and Authentication {2.34-2.36}

    4.2 Access Control {2.37-2.39}

    4.3 Accountability {2.40-2.42}

    4.4 Audit {2.43-2.45}

    4.5 Object Re-use {2.46-2.48}

    4.6 Accuracy {2.49-2.51}

    4.7 Reliability of Service {2.52-2.54}

    4.8 Data Exchange {2.55-2.58}

etc.

Key to the specification style: ○ Informal; □ Semiformal & Informal; ■ Formal & Informal

Figure 6.B.8 Security Target for a System Evaluation

1. Introduction

    1.1 Claimed Rating of Minimum SOM {2.25}

    1.2 Target Evaluation Level {2.26}

2. System Security Policy {2.9-2.15}

    2.1 Security Objectives {2.9}

    2.2 Description of Operational Environment {2.9}

    2.3 Actual Threats {2.9}

3. Model of Security Policy {2.81-2.83}

4. Specification of Security Enforcing Functions {2.18-2.24}

    [Defn of Security Enforcing Functions]

    [Defn of Required Security Mechanisms (optional)]

    4.1 Identification and Authentication {2.34-2.36}

    4.2 Access Control {2.37-2.39}

    4.3 Accountability {2.40-2.42}

    4.4 Audit {2.43-2.45}

    4.5 Object Re-use {2.46-2.48}

    4.6 Accuracy {2.49-2.51}

    4.7 Reliability of Service {2.52-2.54}

    4.8 Data Exchange {2.55-2.58}

    etc.

Key to the specification style: ○ Informal; □ Semiformal & Informal; ■ Formal & Informal

Annex 6.C Effectiveness

Introduction

6.C.1 This annex describes the application of the ITSEC in the area of effectiveness.

Mechanisms

Classifying Mechanisms

6.C.2 This section describes the various types of mechanism which can be used within a TOE.

6.C.3 A security mechanism is defined in ITSEC Paragraph 6.59 as the logic or algorithm that implements a particular security enforcing or security relevant function in hardware and software. A critical mechanism is defined in ITSEC Paragraph 6.22 as a mechanism within a Target of Evaluation whose failure would create a security weakness.

6.C.4 A *type A mechanism* is a security mechanism with a potential vulnerability in its algorithm, principles or properties, whereby the mechanism can be overcome by the use of sufficient resources, expertise and opportunity in the form of a direct attack. An example of a type A mechanism would be an authentication program using a password: if the password can be guessed by attempting all possible passwords in succession, the authentication mechanism is of type A. Type A mechanisms often involve the use of a "secret" such as a password or cryptographic key.

6.C.5 All type A mechanisms in a TOE have a strength, which corresponds to the level of resources, expertise and opportunity required to compromise security by directly attacking the mechanism.

6.C.6 When assessing the strength of a mechanism, the context in which the mechanism operates should be taken into account. See the *Example* subsection below.

6.C.7 A *type B mechanism* is a security mechanism which, if perfectly conceived and implemented, will have no weaknesses. A type B mechanism can be considered to be impregnable to direct attack regardless of the level of resources, expertise and opportunity deployed. A potential example of a type B mechanism would be access control based on access control lists: if perfectly conceived and implemented, this type B mechanism cannot be defeated by direct attack. However, these type B mechanisms can be defeated by indirect attacks which are the subject of other effectiveness analyses.

6.C.8 The security target for a TOE should specify the minimum strength of mechanisms, i.e. the strength of the weakest type A critical mechanism in the TOE. The developer's strength of mechanisms analysis could:

a) identify the critical mechanisms and explain why the remaining mechanisms are not critical;

b) state and confirm the strength of each type A critical mechanism;

c) confirm that type B critical mechanisms have no weaknesses (perhaps by reference to other effectiveness analyses).

Example

6.C.9 A security mechanism is the logic or algorithm that implements a particular security enforcing or security relevant function in hardware and software. For example, a complex password algorithm *A*, might be derived from some simple password algorithm *B*, strengthened by the principle *C*, of limiting the number of re-tries following an authentication failure: one developer might regard this algorithm *A*, as being implemented by a single security mechanism M_A, whereas another developer may choose to regard the same algorithm *A*, as being implemented by two mechanisms M_B, which implements algorithm B, and M_C, which implements principle *C*. Thus, if the existing design only employs algorithm *B* and the two developers both choose to strengthen it by employing algorithm A:

a) the first developer will regard this course of action as strengthening the mechanism;

b) the second developer will regard the same course of action as employing an additional mechanism.

6.C.10 Since both courses of action are in practice identical, the cases of Paragraphs a) and b) above are equivalent. Figure 6.C.1 illustrates the two situations.

6.C.11 In the figure, mechanisms A and B are both type A because they can be defeated by direct attack (e.g. repeatedly guessing passwords). Mechanism A and the mechanism B in the context of mechanism C have identical strength. Mechanism A gains some of its strength from the fact that it includes a limit on the number of retries. Mechanism B does not include this limit, but it has to be evaluated taking its context into account. Therefore, the fact that mechanism C limits retries, strengthens the combination of mechanisms B and C.

The Effectiveness Criteria

Effectiveness and Correctness

6.C.12 In general, the division of work between correctness and effectiveness depends on the security target. This is because correctness applies to functionality and is measured against what is specified in the security target, whereas effectiveness is concerned with a lack of exploitable vulnerabilities. The more detail is included in the security target, the greater is the proportion of evaluation effort concerned with correctness assessment.

6.C.13 For example, a security target might require identification and authentication functionality to be implemented with a high strength of mechanism, but it does not specify the mechanism. An implementation which allowed users to have two-character passwords would then be rejected as ineffective. If the security target disallows two-character passwords, the same implementation would fail for correctness reasons.

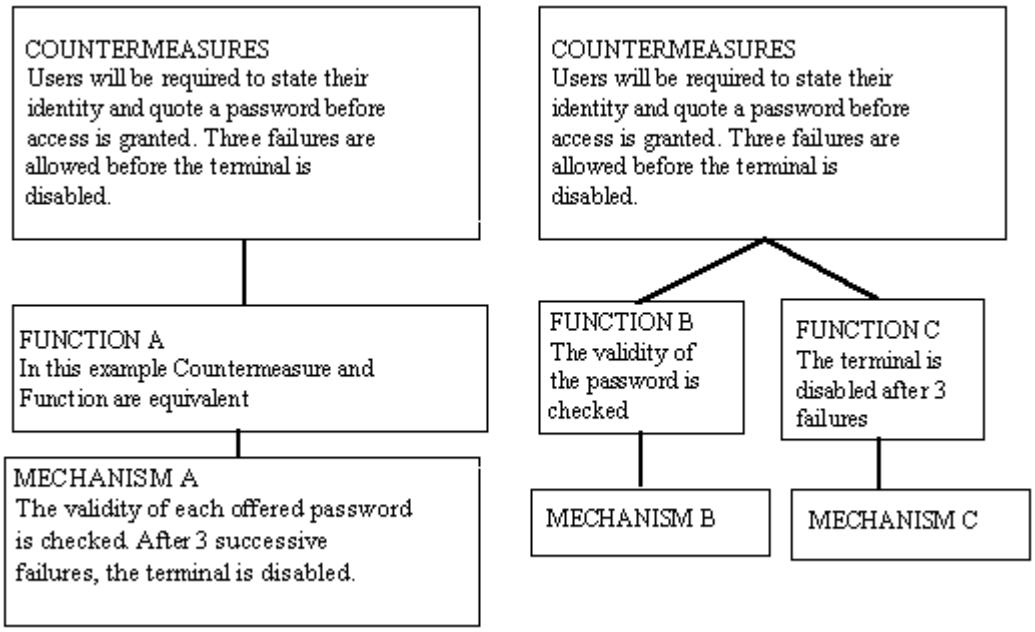


Figure 6.C.1 Two ways of Treating Mechanisms

Aspects of Effectiveness

6.C.14 This subsection considers the relationships between the effectiveness criteria.

6.C.15 It is constructive to consider the effectiveness criteria from the point of view of a developer. The developer should perform a risk assessment to determine the required security enforcing functions, based on the following:

a) a broad definition of the required (non security relevant) functionality of the TOE;

b) the threats to the TOE and/or the security objectives of the TOE;

c) the assets to be protected by the TOE (assets can be information or software whose confidentiality, integrity and availability are to be protected).

6.C.16 In selecting these security enforcing functions, the developer should decide whether they are:

a) suitable, in that they should counter the threat(s);

b) able to work together, in the case where more than one security enforcing function has been chosen (i.e. bind together) in a way that is mutually supportive and provides an integrated and effective whole.

6.C.17 A simple, schematic, illustration of this process is provided in figure 6.C.2. This figure (and figures 6.C.3 and 6.C.4) represent:

a) the assets by an "ECU" sign;

b) the threats by "nails" with a length "proportional" to the level of expertise, opportunity and resource available to the attacker;

c) the countermeasures (e.g. security enforcing functions) by a "wall" which has a thickness "proportional" to the strength of the type A mechanisms which implement it (i.e., the countermeasure's ability to defend against direct attack).

6.C.18 The longer the "nail", the greater is the severity of the threat; the thicker the wall, the greater is the ability of the countermeasure to defend the assets against that threat. Indeed, the TOE is deemed to be secure if the assets are completely surrounded by a wall which has a minimum thickness equal to or greater than the length of any nail.

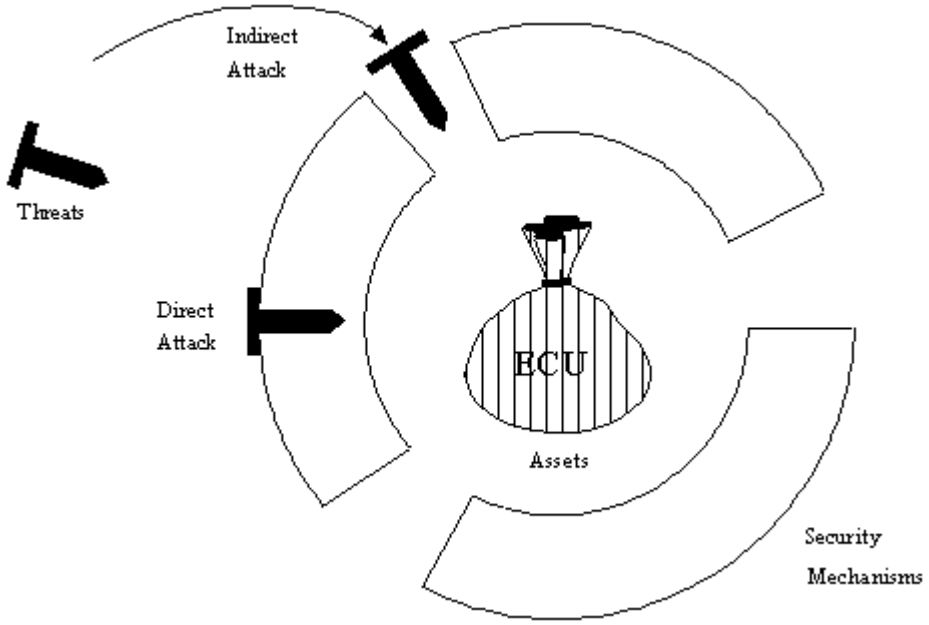


Figure 6.C.2 The Failure of Suitability and Binding

6.C.19 Figure 6.C.2 illustrates the case where the selected security enforcing functions are insufficient to counter the threat, even though their mechanisms are of sufficient strength. The figure illustrates the design of a secure operating system (e.g. F-B2) where the developer has neglected to include the necessary mechanisms to protect the traditional security enforcing functions (e.g. Identification and Authentication, Access Control etc.) from external interference and tampering. Indeed, at this stage of vulnerability assessment, the developer could argue that his solution is so far:

a) unsuitable, because it will not counter the threat;

b) does not bind, because it does not form an integrated whole.

6.C.20 These defects are overcome in figure 6.C.3, which shows the introduction of a second set of countermeasures which have the objective of protecting the first set from external interference or tampering.

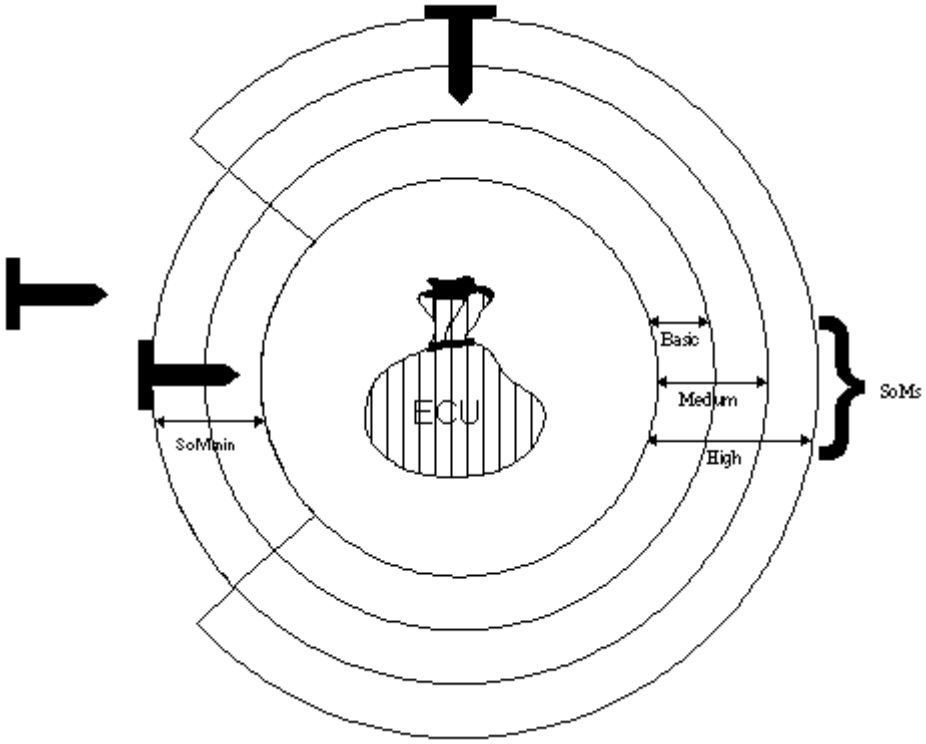


Figure 6.C.3 A Secure TOE

6.C.21 This figure also indicates the countermeasure `wall' thicknesses necessary to meet the ITSEC Strength of Mechanisms (SoM) definitions: *basic*, *medium* and *high*. The protection mechanisms are shown as *high* and the original countermeasures as *medium*. If the minimum SoM (SoMmin) is *medium*, then this figure indicates that the TOE should meet the suitability, binding and SoM criteria:

a) the minimum thickness of the `wall' is medium, thereby meeting the SoM requirement;

b) the `wall' completely surrounds the assets and there are no gaps, therefore the countermeasures counter the threat and the suitability and binding criteria are met.

6.C.22 At each stage of development, the developer should repeat his vulnerability assessment. Indeed, from an evaluation point of view, he should do this until all the information cited in ITSEC figure 4, has been considered for the evaluation level in question.

6.C.23 In figure 6.C.4(a), a vulnerability is shown as a `thinning of the countermeasure wall', i.e. a failure of the wall to meet the thickness required by SoMmin. In this case, the SoM rating of the TOE is the lowest of the ratings for each critical mechanism.

6.C.24 In accordance with the ITSEC, there are four ways in which the developer may modify, or progress, his design in order to counter this vulnerability:

a) The developer is at liberty to regard the vulnerability as a failure of the underlying algorithms, principles and properties of the mechanisms concerned to meet SoMmin. In this case, the developer's recourse is to modify the existing algorithms, principles and properties (or adopt new algorithms, principles and properties) which do meet the required SoMmin. If the developer elects to take this course of action, the result will be as depicted in figure 6.C.4(b).

b) Alternatively, following ITSEC Paragraph 3.27 (first item) of the **Construction Vulnerability** Assessment criterion, the developer can introduce some additional security mechanism (or mechanisms) internal to the TOE. If the developer elects to take this course of action, the result will be again as depicted in figure 6.C.4(b). Indeed, this course of action is synonymous with the first.

c) Following the second item of ITSEC Paragraph 3.27 of the Construction Vulnerability Assessment criterion (or the **Operational Vulnerability** criterion), the developer can require the introduction of some external countermeasure. For this course of action, the result will be as depicted in figure 6.C.4(c). If this course is followed, the countermeasure should be documented in the security target.

d) Finally, following the Ease of Use criterion, the developer could introduce a combination of internal and external measures which although they do not have the effect of countering the vulnerability directly (i.e. by thickening the wall), do have the effect of bringing any attempt to exploit the vulnerability to the attention of an end-user or an administrator. This course of action is depicted in figure 6.C.4(d), and would be the case, for example, where the use of covert channels, which cannot be removed, is monitored by a secure operating system.

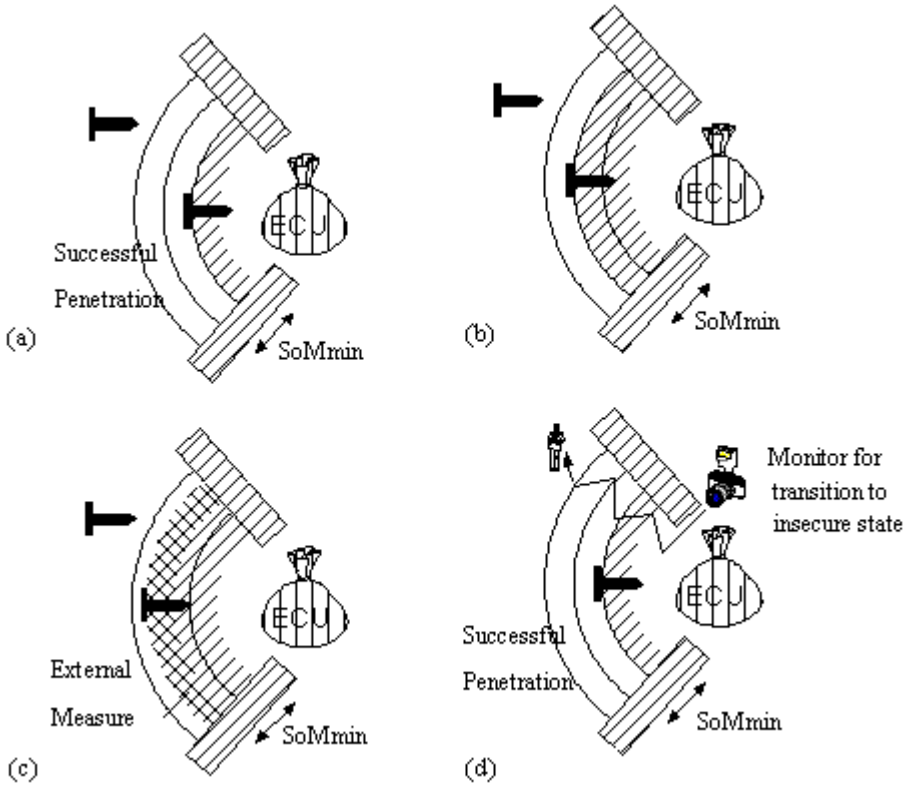


Figure 6.C.4 Resolving Security Vulnerabilities

6.C.25 Once the developer has chosen how he will counter the previously identified vulnerabilities, application of the Binding of Functionality criterion will check that the solution does indeed solve the vulnerability problem (as identified at this level of design) and does not introduce any further vulnerability. If a further vulnerability were to be discovered, then the developer would have to re-trace his steps, selecting different internal and external countermeasure solutions until the Binding of Functionality criterion is at last satisfied.

6.C.26 It should be noted that when a vulnerability is found, it will sometimes be possible to characterise it in more than one way; for instance, it may be difficult to decide whether it is a suitability or a binding problem. In practice, this is not a problem. It is more important to have confidence that all vulnerabilities have been found than to be easily able to distinguish between different types of vulnerability.

6.C.27 To summarise, effectiveness errors can be of two types:

a) Errors in content, presentation and evidence for a particular effectiveness deliverable. This type of error will correspond to a particular effectiveness aspect;

b) Vulnerabilities found during penetration testing. It may be more difficult to assign an effectiveness aspect to this type of vulnerability.

Estimating Strength of Mechanisms

6.C.28 According to the ITSEC (Paragraphs 3.6-3.8) the meaning of strength of mechanisms ratings is as follows:

a) For the minimum strength of a critical mechanism to be rated *basic* it shall be evident that it provides protection against random accidental subversion, although it may be capable of being defeated by knowledgeable attackers.

b) For the minimum strength of a critical mechanism to be rated *medium* it shall be evident that it provides protection against attackers with limited opportunities or resources.

c) For the minimum strength of a critical mechanism to be rated *high* it shall be evident that it could only be defeated by attackers possessing a high level of expertise, opportunity and resources, successful attack being judged to be beyond normal practicability.

6.C.29 These definitions are informal, intended to be meaningful to users of a TOE. This subsection gives guidance on more objective means of measurement.

6.C.30 Since strength of mechanisms concerns expertise, opportunity and resources, it is necessary to expand on the meaning of these terms:

a) *Expertise* concerns the knowledge required for persons to be able to attack a TOE. A *layman* is someone with no particular expertise; a *proficient* person is someone familiar with the internal workings of the TOE, and an *expert* is someone familiar with the underlying principles and algorithms involved in the TOE.

b) *Resources* concern the resources an attacker must expend to successfully attack the TOE. Evaluators are usually concerned with two types of resources: *time* and *equipment*. Time is the time taken by an attacker to perform an attack, not including study time. Equipment includes computers, electronic devices, hardware tools, and computer software. For the purposes of this discussion,

- *In minutes* means an attack can succeed in under ten minutes; *in days* means an attack can succeed in less than a month, and *in months* means a successful attack requires at least a month.

- *Unaided* means no special equipment is required to effect an attack; *domestic equipment* is equipment which is readily available within the operational environment of the TOE, or is a part of the TOE itself, or can be purchased by the public; t*special equipment* is special-purpose equipment for carrying out an attack.

c) *Opportunity* covers factors which would generally be considered outside an attacker's control, such as whether another person's assistance is required (collusion), the likelihood of some specific combination of circumstances arising (chance), and the likelihood and consequences of an attacker being caught (detection). These factors are difficult to rate in the general case. The case of collusion is covered here, but other factors may have to be considered. The following forms of *collusion* are discussed: *alone* if no collusion is required; *with a user* if collusion is required between an attacker and an untrusted user of the TOE for an attack to succeed; and *with an administrator* if collusion with a highly trusted user of the TOE is required. This definition of collusion presumes that the attacker is not an authorised user of the TOE.

6.C.31 The factors discussed above are not believed to be final or complete; they are intended for guidance only. Once the factors have been evaluated for a particular mechanism, the following rules can be used to calculate the mechanism's strength:

a) If the mechanism can be defeated by a layman in minutes unaided, then it does not achieve *basic*.

b) If the mechanism can only be defeated by an expert using special equipment, taking months, and requires collusion with an administrator, then the mechanism achieves *high*.

c) If the mechanism can only be defeated by collusion with a user, the mechanism achieves at least *medium*.

d) If the mechanism can only be defeated by collusion with an administrator, then the mechanism achieves at least *high*.

e) If a successful attack requires months, the mechanism achieves at least *medium*.

f) If a successful attack requires an expert, months of effort and special equipment then the mechanism rates *high*, regardless of whether collusion is required.

g) If successful attack requires days of effort then the mechanism must be rated at least *basic*.

h) If successful attack can be performed in minutes by anyone other than a layman then the mechanism is *basic*.

i) If successful attack requires an expert using domestic equipment over days then the mechanism is *medium*.

6.C.32 Rather than compute these predicates, however, the evaluators may make use of the guidelines provided in figures 6.C.5 and 6.C.6. Add together the two numbers found by looking up TIME and COLLUSION in figure 6.C.5 and by looking up EXPERTISE and EQUIPMENT in figure 6.C.6:

a) If the result is 1 then the strength is not even *basic*..

b) If the result is greater than 1 but no higher than 12 then the strength is *basic*.

c) If the result is greater than 12 but no higher than 24 then the strength is *medium*..

d) If the result is greater than 24 then the strength is *high*.

6.C.33 The values contained within figures 6.C.5 and 6.C.6 only serve the purpose of evaluating the predicate. There is no other significance. For example a layman unaided within minutes, with a user (value 13) is no better or worse than an expert unaided taking months/years alone (value 22) - both are rated *medium*.

Figure 6.C.5 Table for Time and Collusion

	COLLUSION		
TIME	alone	with a user	with an administrator
within minutes	0	12	24
Within days	5	12	24
Months/years	16	16	24

Figure 6.C.6 Table for Expertise and Equipment

	EQUIPMENT		
EXPERTISE	unaided	Using domestic equipment	using specialist equipment
layman	1	n/a	n/a
proficient	4	4	n/a
expert	6	8	12

6.C.34 These tables should be used as guidance since they may not be applicable to all mechanisms and operational environments. These tables are not to be used to rate cryptographic mechanisms (see ITSEC Paragraph 3.23).

Annex 6.D Impact Analysis for Re-evaluation

Introduction

6.D.1 It is unrealistic to assume that a TOE itself, its operational environment or its development environment will not change over time. It is more likely that the sequence of the processes in the IT security framework is permanently ongoing, as indicated in part 1 of the ITSEM, figure 1.1.1.

6.D.2 An evaluation result applies only for a specific release or version of an IT system or IT product. Therefore any change to the TOE or its deliverables could make a re-evaluation necessary. A complete evaluation each time a change occurs is unnecessary and it is possible to take advantage of previous evaluation results. As the impact of a change is not always security relevant, a process called impact analysis is required to indicate the consequences of a change in the above mentioned areas, i.e. whether a change results in a need for re-evaluation.

6.D.3 This annex gives basic guidance to sponsors, developers and system accreditors by describing the impact analysis process which involves the following issues:

a) how to establish the necessity of a re-evaluation;

b) how to identify the parts of the TOE affected;

c) which evaluator actions have to be repeated.

6.D.4 Part 3 of the ITSEM (describing *philosophy, concepts* and *principles*) and part 4 of the ITSEM (describing the *evaluation process*) are the basis for evaluation. Part 4, chapter 4.6, covering *re-use*, is closely related to this annex.

Impact Analysis

Overview

6.D.5 An evaluation result only applies to the release and version of a TOE that was evaluated. If a TOE, its operational environment, or its development environment is subsequently changed, it is the responsibility of the sponsor to determine the change type and the consequences for the certificate/certification report.

6.D.6 According to the change type it may be necessary for the sponsor/developer to notify the certification body of the change. If a re-evaluation is required, it will be necessary for the sponsor/developer to supply relevant deliverables to an ITSEF.

6.D.7 The main rule for this process is that all decisions have to be made according to the evaluation level which was awarded to the TOE. Since the awarded evaluation level is a measure of the confidence that can be held in a TOE fulfilling its security objectives, it is necessary for a change to be scrutinised with the same degree of rigour as during the initial evaluation. Otherwise the level of confidence cannot be maintained.

6.D.8 A special case exists for changes in the development environment or the hardware platform. During the course of an evaluation, tools will be identified which are security relevant, e.g. the compiler used to create the object code.

6.D.9 Since the development tools used and their influence on the confidence held in the TOE vary widely, no universally applicable scheme for their treatment can be developed. Therefore changes in the development environment or the hardware platform have to be dealt with on a case by case basis. It is the task of the evaluation to identify those tools which are security relevant with respect to the target evaluation level. If requested these can be recorded in Chapter 7 of the ETR, together with other useful information for impact analysis.

Pre-Requisites

6.D.10 The impact analysis process requires information about the components of the TOE and its development environment. If a sponsor believes re-evaluation is likely to be necessary for a TOE, he should request the ITSEF to record such information in Chapter 7 of the ETR.

6.D.11 A TOE and its development environment consists of a collection of components working together each component having one of the following properties. The component types are defined as follows (see part 3 of the ITSEM):

a) SE: Security Enforcing;

b) SR: Security Relevant;

c) SI: Security Irrelevant.

6.D.12 The following examples apply only for confidentiality requirements. The identification and authentication component of an operating system implements a security enforcing function and is therefore of the type SE. The scheduler of an operating system is security relevant software and therefore of the type SR. The memory management unit together with the data it operates on implemented in firmware and hardware is security enforcing and therefore of the type SE. The CPU of a machine, again a combination of firmware and hardware, is security relevant and therefore of the type SR. Unprivileged user programs are of the type SI.

The Process

6.D.13 The concept of the impact analysis process is presented in figure 6.D.1. It consists of two steps, the first determines the change type, and the second leads to the decision about the necessity of re-evaluation and to the required actions depending on this change type.

Step 1 (Determine change type)

6.D.14 The type of change made to the TOE has to be determined by means of figure 6.D.2. A detailed description of the process and the rationale follows:

6.D.15 A change could be related to the security target, the effectiveness criteria or the correctness criteria. This change could have a consequence on the confidence that can be held in a TOE.

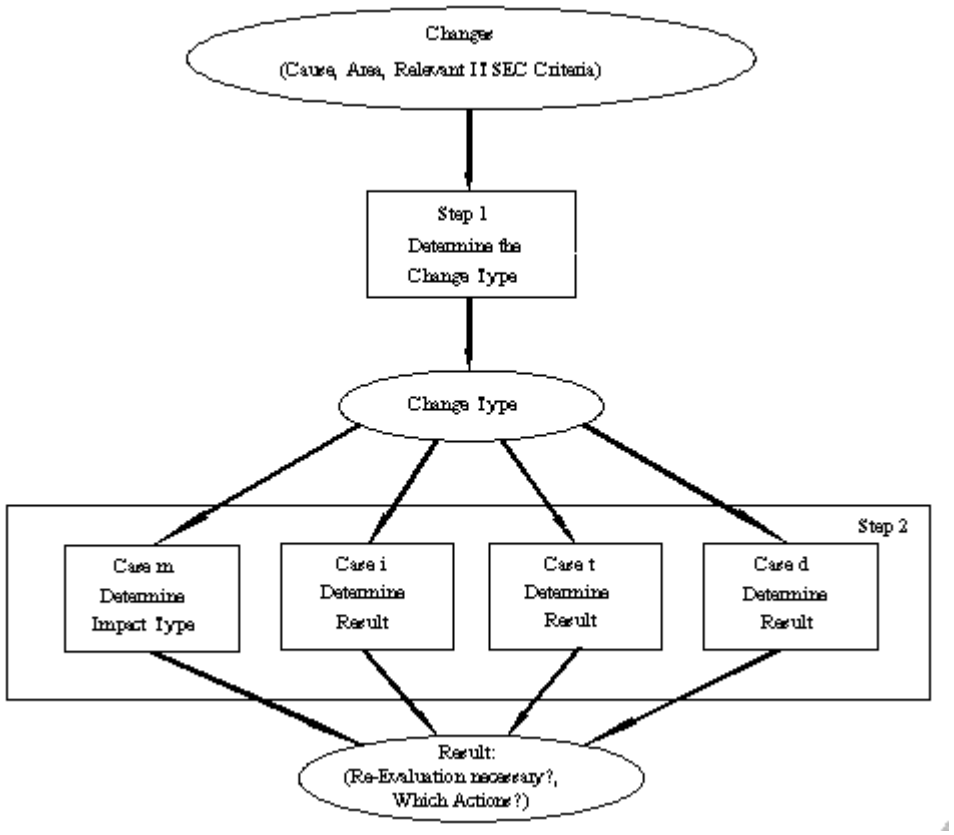


Figure 6.D.1 Overview of the Impact Analysis Process

6.D.16 The following figure shows possible causes and impacts of a change to a TOE. The figure has four columns: the cause of the change, the area to which the change is related, the relevant criteria of the ITSEC affected by the change and the resulting *change type*.

6.D.17 The change type has four possible values:

m:. for a change ultimately leading to a *modification* of the TOE;

i:. for a change which has only *indirect* effects on the TOE;

d:. for a change to *documentation* which could have a consequence on the operation of the TOE;

t:. for a change to a *tool* used during development.

Cause	Area	Relevant ITSEC Criteria		Change Type
New threat added New security enforcing function added Changed SoM rating	Security Target	Phase 1 - Requirements Phase 2 - Architectural Design Phase 3 - Detailed Design Phase 4 - Implementation	and/or and/or and/or	m m m m
Exploitable vulnerability found	Effectiveness	Phase 1 - Requirements Phase 2 - Architectural Design Phase 3 - Detailed Design Phase 4 - Implementation	and/or and/or and/or	m m m m
Change of development process	Correctness	Phase 1 - Requirements Phase 2 - Architectural Design Phase 3 - Detailed Design Phase 4 – Implementation	and/or and/or and/or	m m m m
Change of development environment		Aspect 1 - Configuration Control Aspect 2 - Prog. Languages and Compiler Aspect 3 - Developers Security		i t i
Change of operational documentation		Aspect 1 - User Documentation Aspect 2 - Administration Docn		D d
Change of operational environment		Aspect 1 - Delivery and Configuration Aspect 2 - Startup and Operation		d d

Figure 6.D.2 Change Types To A TOE

Step 2 (Determine Result)

6.D.18 This step determines the result, i.e. whether a re-evaluation is necessary and which actions are required. It is divided into four different cases which are to be performed depending on the change type.

6.D.19 This section presumes that the component types are unchanged. This is not always true. For example, on one re-evaluation some components which were previously *security relevant* may become *security enforcing*, and vice versa. Also, if there is a change in architecture, the separation between security enforcing, security relevant, or security irrelevant components may change.

Case m (Determine Result for change type "m")

6.D.20 The change type m is subdivided into four different *sub change types* which are defined as follows:

m0: A change to the security target occurs.

m1: At the architectural design level a change occurs which does not affect the security target.

m2: At the detailed design level an isolated change occurs. This change is not visible at the architectural design level and therefore no update of the architectural design level documentation is necessary.

m3: At the implementation level an isolated change occurs. This change is not visible at the detailed design level, therefore no update of the detailed design level documentation is necessary.

6.D.21 The following additional constraints have a consequence on the change sub types. If either for m2 or m3 a change cannot be identified to be 'isolated', e.g. it spreads over a number of basic components, the next higher type of change applies. So a change at the implementation level (m3) which does not have the property 'isolated' is equivalent to a type m2 change. The same rule applies to type m2 changes, they become a type m1 change. The evidence that a change is of type m0, m1, m2, or m3 has to be supplied by the sponsor/developer.

6.D.22 The impact type of the changes is determined using the appropriate table for the target evaluation level in figure 6.D.3. Each of the five defined impact types leads to a different result.

6.D.23 The following items have to be identified as input for this case:

a) the evaluation level the TOE has reached;

b) the sub change type (m0 to m3);

c) the component type (SE, SR or SI) of the changed component(s).

6.D.24 The output from the table is the impact type for this specific change. The impact type gives information on the actions to be performed by the sponsor/developer and the ITSEF.

Impact Types

6.D.25 The entries in the tables of figure 6.D.3 distinguish five possible values (the impact types I1 - I5) which show the consequences of a change to the evaluated TOE. It must be noted that the impact type may change after a thorough analysis. Impact types are summarised in figure 6.D.4.

Evaluation Level	Sub-change type
Component Type	Impact Type

E1	m0	m1	m2	m3
SE	I5	I4	I2	I2
SR	I5	I3	I2	I2
SI	X	I1	I1	I1

E2	m0	m1	m2	m3
SE	I5	I4	I3	I2
SR	I5	I3	I3	I2
SI	X	I1	I1	I1

E3	m0	m1	m2	m3
SE	I5	I4	I4	I3
SR	I5	I4	I3	I3
SI	X	I1	I1	I1

E4	m0	m1	m2	m3
SE	I5	I5	I5	I4
SR	I5	I4	I3	I3
SI	X	I1	I1	I1

E5	m0	m1	m2	m3
SE	I5	I5	I5	I4
SR	I5	I5	I4	I3
SI	X	I1	I1	I1

E6	m0	m1	m2	m3
SE	I5	I5	I5	I5
SR	I5	I5	I5	I4
SI	X	I1	I1	I1

"X" reperesents an impossible combination

Figure 6.D.3 Impact Types For E1 To E6

Figure 6.D.4 Summary of Impact types	
Impact Type	Required Actions
I1	Inform certification body
I2	I1 + provide test documentation to certification body
I3	provide deliverables to ITSEF ITSEF checks deliverables for content presentation and evidence
I4	I3 + ITSEF performs all ITSEC evaluator actions
I5	Full re-evaluation

Impact Type I1

6.D.26 Changes which result in an impact type of I1 are security irrelevant and no action is required except when this change has editorial consequences to the evaluation result and certificate/certification report. Nevertheless the sponsor/developer should inform the certification body if there is any doubt.

Impact Type I2

6.D.27 The certification body is informed about the change since there may be an impact on the enforcement of the security policy. Test documents are sent to the certification body together with a change notice appropriate to the respective evaluation level.

Impact Type I3

6.D.28 The certification body is informed about the change, as changes of this type may also impact the enforcement of the security policy. The information accompanying the change notice has to fulfil the requirements for content, presentation, and evidence of the respective evaluation level and the relevant phase(s) or aspect(s). An ITSEF checks that the information provided meets all requirements for content, presentation, and evidence both for effectiveness and correctness. Evidence for effectiveness will not be required for changes of the m2 type in the table for E2 and changes of the m3 type in the table for E3 (since they are not required in the original evaluation).

Impact Type I4

6.D.29 Changes which result in an impact type of I4 most probably affect the enforcement of the security policy. The same rules as in I3 apply, but the information supplied according to I3 is not sufficient to demonstrate that after the checks have been performed, the evaluation level is still valid. An ITSEF has to perform activities as identified in the evaluator actions of the ITSEC with respect to the evaluation level and phase for both correctness and effectiveness. Should, during the course of this activity, a correctness error or an exploitable vulnerability be found, the result type changes to I5.

Impact Type I5

6.D.30 Changes which result in an impact type of I5 are always security relevant. The certification body is informed about the change. The information accompanying the change notice has to be sufficient for a meeting between the certification body, ITSEF and sponsor to be convened. In this meeting the scope of the re-evaluation necessary will be discussed. The certification body must be notified if a problem which gave rise to the proposed change. This notification should include an updated list of vulnerabilities.

Change Notices

6.D.31 The information to be supplied with a change notice varies, but for I1 to I4 the following information is required:

a) evidence that the change type is as claimed;

b) evidence that the component type is as claimed;

c) evidence appropriate to the new target evaluation level.

6.D.32 For example, evidence for item c) consists of the relevant deliverables with the changes identified together with an explanation why the criteria 'isolated' and 'not visible at the next design level' are met.

Case i (Determine Result for change type "i")

6.D.33 For this change type it has to be checked with regard to the target evaluation level whether the requirements are still valid after the change.

Case d (Determine Result for change type "d")

6.D.34 It has to be checked whether the change in the documentation could have an influence on the *correctness - operation, ease of use or operational vulnerability* criteria. If these are not affected, no further actions are required. If they are affected they have to be applied again.

Case t (Determine Result for change type "t")

6.D.35 For this change type the evaluation result remains valid for a target evaluation level of E2 or lower. For a target evaluation level of E3 and higher it has to be checked with regard to the *programming languages and compilers criteria* whether the actual change can invalidate the confidence level held in the TOE.

The Re-Evaluation Process

6.D.36 After the decision on the necessity for a re-evaluation and the actions required, the actual re-evaluation is performed.

6.D.37 Since the evaluation level is a measure of the confidence that can be held in a TOE to fulfil its security objectives, it is necessary for a re-evaluation to be performed with the same degree of rigour as the initial evaluation, or with an even higher degree of rigour if the target evaluation level of the re-evaluation is higher. Otherwise the level of confidence cannot be upheld.

6.D.38 Note that there may have been improvements in the development procedures since the original evaluation (e.g. following problem reports). These changes could affect the level of work that the re-evaluators consider necessary for this or future re-evaluations.

Annex 6.E Guidance for Tool Providers: Building an Evaluation Workbench

Introduction

6.E.1 This annex describes various ideas for the specification and construction of an evaluation workbench. Tool providers or vendors should consider these concepts during their building of evaluation tools.

6.E.2 The basic ideas come from the software development engineering world. Computer Aided Software Engineering (CASE) provides developers with a set of tools implementing methods used to specify, design, program, test and validate software. An Integrated Project Support Environment (IPSE) is a software platform where a developer can place every tool needed to cover the complete development cycle. With addition of tools the IPSE becomes a Populated IPSE (PIPSE). A workbench built on a PIPSE may offer a development methodology (e.g. organisation) and a range of management tools to produce quality software.

6.E.3 The concept of this annex is to adapt these principles to the building of an Evaluation Workbench which will provide evaluators with tools to perform their work efficiently and to ensure that the evaluation principles of part 3 of the ITSEM are adhered to. The greatest challenge remains the variety of the evaluations conductable under ITSEC and the difficulty in finding common solutions to cover the whole IT field. This should be resolved by the ever greater tendency towards standardisation and openness in the field.

6.E.4 The goal of this annex is therefore:

a) To present the basic concepts for building an evaluation PIPSE in order to show that benefit can be obtained from a close link between techniques and tools (described in part 4, chapter 4.5) to conduct evaluations.

b) To give the typical characteristics expected for the tools to populate the evaluator's workbench. No attempt is made here to address the specific tools and techniques used during the development of systems. Clearly, the greater commonality between development tools and evaluation tools, the cheaper the evaluation workbench will be.

c) To provide additional information on categories of tools where the selection of tools by the evaluators is difficult; the content of that section may evolve as IT evolves..

A PIPSE for the Evaluation Workbench

Concept

6.E.5 It is beneficial if the techniques and tools described in part 4, chapter 4.5 work together to support the whole evaluation process. To achieve this they can be organised within an IPSE. The IPSE provides an infrastructure (defined formats for data exchange, a shared evaluation database, common services for word processing, manipulation and display of results etc.) within which the individual tools may be integrated.

6.E.6 The PIPSE addresses the major problems of evaluation productivity, duration and quality.

6.E.7 In compliance with international and European standards and approaches e.g. [ECMA], it is also important to aim at an Open System approach and to speak of an Open IPSE or Open PIPSE for evaluation. Finally it is useful to include hardware CAD or other TOE-specific tools..

Benefits

6.E.8 The advantages of a PIPSE are that it facilitates:

a) evaluation project management: cost and timing estimation, evaluation project planning, scheduling of the evaluation activities etc.;

b) evaluation configuration management: the evaluation activities should be conducted under configuration management (particularly important in the case of correction of errors found during evaluation and also in the case of re-evaluation after modification of the TOE);

c) evaluation documentation production and management;

d) inter-PIPSE communication tools: it may be useful to link together evaluators who are working on the same TOE including, if security considerations allow, two or more ITSEFs;

e) the creation of an evaluation database: care is required when collecting and storing proprietary information from vendors;

f) other associated services: e.g. on-line help.

6.E.9 The PIPSE will also benefit from the integration of new tools to permit the automation of the evaluation process.

Architecture

6.E.10 Figure 6.E.1 shows a possible general layered architecture, which may be used to develop the evaluation PIPSE. The architecture includes:

a) an operating system which provides some basic security mechanisms used to protect the proprietary information evaluated (e.g. source code) and the information produced during the evaluation process;

b) a layer of common services providing support for an integrated software development environment (possibly common with the development environment for the TOE) providing the basis to integrate tools e.g. [PCTE];

c) a software backplane enforcing homogeneity of methods and tools and, if necessary, evaluation rules;

d) a horizontal environment providing basic management services like documentation editing and management functions, a configuration management tool, project management, electronic mail; this horizontal environment may also receive the developer's tools necessary for evaluation (compilers, libraries);

e) a vertical environment where the different evaluation tools (as described in part 4, chapter 4.5) will be inserted;

f) a user-friendly man-machine interface.

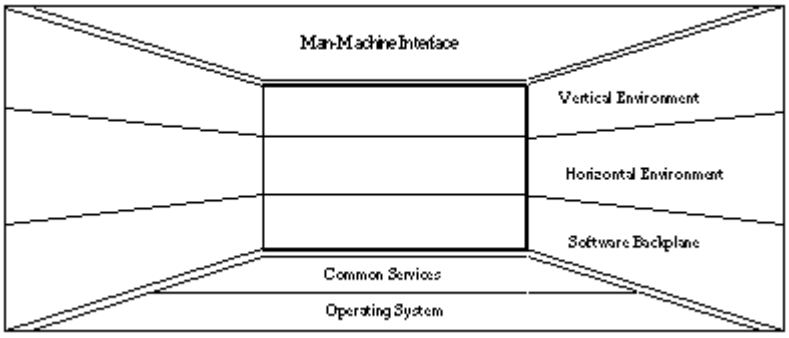


Figure 6.E.1 Possible PIPSE Architecture

Checklists

6.E.11 It is required to ensure that ITSEFs consider all relevant facts during the application of each ITSEC criterion. The ITSEC does not provide a general purpose checklist because the specific evaluator actions will be governed by the nature of the TOE and its security requirements as defined in its security target.

6.E.12 The workbench can aid in generating and validating consistent checklists across different evaluations so that if two ITSEFs were to evaluate the same TOE to the same security target they would generate the same checklists.

Populating an Evaluation Workbench

General

6.E.13 This section describes the desirable features of evaluation tools. It concentrates mainly on the characteristics of the tools which aid in achieving the principles of ***repeatability, reproducibility*** and ***objectivity***.

Technical Suitability of Tools

6.E.14 The technical suitability of a tool can be characterised by its range of application and the degree of accuracy obtained.

6.E.15 *Range of application*: there are two possible approaches to the generation of tools for the evaluation activities. The first is to attempt to create tools which are as universal as possible; the second is to attempt to create a set of specialised tools.

6.E.16 The specialised tools have the disadvantage of limited applicability and a lack of flexibility. Thus it may be difficult to use specialised tools effectively. With the increasing standardisation of IT and the better definition of the evaluator's tasks the area of specialist tools will be of greater value.

6.E.17 The ability to combine tools is especially important for evaluation in areas where there is a lack of a single suitable tool. The PIPSE organisation allows efficient and flexible combination.

6.E.18 *Degree of formalisation*: as the level of evaluation rises, the tools must be able to support the semiformal and formal methods as required by the ITSEC. The tools' properties may be characterised as follows:

a) the input languages have a well-defined syntax and semantics;

b) a mathematical theory or a formal model exists which ensures the validity of the results produced.

Ease of Learning and Use of Tools

6.E.19 The underlying technique on which the tool is based should be easy to learn and easy to use so as to preclude mistakes and false reasoning. This does not imply simplicity of the technique itself. Even where techniques are complex or have a complex underlying theory, evaluators should be able to use the tool effectively. Training is a key issue here.

6.E.20 *Ease of learning* is a basic requirement. Even if the task being automated is complex, a tool should permit useful results to be obtained. Factors which affect ease of learning are:

a) quality and relevance of documentation (including error messages and on-line help);

b) quality of training;

c) design of the man-machine interface;

d) adherence to standards.

6.E.21 The documentation should be complete, including guidance and examples of use in evaluation.

6.E.22 *Ease of preparation of input* is desirable for the tool to be as flexible as possible in accepting variations in the format of inputs.

6.E.23 *Ease of interaction* improves the effectiveness of evaluators. Consideration of the following will assist interaction, and hence will increase the tool's suitability:

a) screen display;

b) command structure (i.e. menus, prompts) and how can they best be named.

Requirements on Outputs for Tools

6.E.24 *Suitability of output*: the suitability of a tool is enhanced by the relevance and clarity of its output. The ease of interpretation of the output of a tool is influenced by the same factors that influence the ease of preparation of input and the ease of interaction. The clarity of the output can have a significant impact on the usefulness of the tool.

6.E.25 Whatever the types of output, they should be well presented. The overall results should be provided in a straightforward manner, presenting precise conclusions.

6.E.26 Output requirements will be discussed under the following headings: *recording of evaluator activities, positive findings* and *validity of results*.

6.E.27 *Recording of evaluator activities*: the requirement for recording evaluator activities is particularly important in the case of using interactive tools. A means of repeating the application of the tool is required so that a result can be demonstrated to other evaluators or to other parties. One way of achieving this is for the tool to record every command sequence entered so that these can be re-entered manually or, preferably, be automatically recalled (this may be a standardised mechanism of the workbench itself). A detailed record of evaluator activities is one of the prime advantages of automating the evaluation process.

6.E.28 *Positive findings*: if the output from the tool is a statement that a pertinent result holds, then this is a clear advantage over the tool which simply produces no negative indication.

6.E.29 *Lack of negative findings*: it is important that if a tool is designed to search for certain undesirable features, their absence is easy to spot. The absence of negative findings provides evidence to the evaluation process even if it does not guarantee a lack of vulnerabilities but will be difficult to spot if the output is not clear.

6.E.30 *Validity of results*: tool output should be trustworthy; if the evidence is from a formal tool, this trustworthiness can be stated in terms of soundness. That is, if the tool can only prove 'true' results one should have more confidence than if it also can prove 'false' results.

Commercial Viability of Tools

6.E.31 Finally it is desirable to use only tools which have good commercial properties; these include portability, maintenance and enhancement.

6.E.32 *Portability*: the portability of a tool refers to whether the tool is available for different operating systems and types of hardware. In evaluation, because of the variety of operating systems that are worked on, portability is a major advantage. However, the question "will a port produce an identical tool ?" has to be addressed carefully.

6.E.33 *Maintainability*: it is important that the tool remains useable if the operating system on which it runs is upgraded. This is really a requirement on the tool's provider to continue to maintain it.

6.E.34 *Enhancement*: tools evolve with the techniques they implement. Improved functionality can be added but, for reasons of repeatability, changing a tool should not change the applicability of results produced previously with that tool.

Annex 6.F Model for Composition and Example Application

Purpose

6.F.1 This annex is addressed to sponsors, system integrators and system accreditors, who are concerned with the composition of previously evaluated TOEs.

6.F.2 The purpose of this annex is to:

a) describe a model for composition of previously evaluated TOEs;

b) describe by way of example how to use the model.

6.F.3 Due to the complexity of the general case only a simplified model can be described. Further guidance on this matter should be obtained from the certification bodies.

Summary

6.F.4 This annex begins by describing a simple model of a component and then shows how the model can be used to describe the composition of two previously evaluated components.

6.F.5 Since there are a number of composition possibilities, two cases of composition are given to show the relevant properties of composition.

6.F.6 Practical experience in the composition of evaluated products is limited.

The Model for Composition

6.F.7 In the context of composition, the term *component* is used in this annex to refer to a pre-evaluated component used in the construction of a TOE. The use of this term is in accordance with the definition in the ITSEC (i.e. a component is an identifiable and self-contained portion of a TOE) since the result of the composition is itself a TOE.

6.F.8 A component is seen as a "white box" (indicating that its internal details are known to a certain degree, which depends on the evaluation level), in contrast to a "black box", where the internal details are not known.

6.F.9 A component in the model is described by:

a) a set of predicates P which it upholds;

b) an interface which delivers services to the environment in which the component resides and an interface for the provision of services to the component;

c) assumptions about the environment in which the component resides;

d) its internal details.

6.F.10 The set of predicates P is directly related to the functionality of the component. This functionality can either be security enforcing (it is related to the security target) or it is security relevant (e.g. it is there to support a security enforcing function). The set of predicates can range from a complete security policy to a single predicate describing one necessary property of a component.

6.F.11 The services which the interface delivers are either to be used by another component or delivered to a user. This interface can be called a producer interface. The level of detail of the interface descriptions and the descriptions of the services it delivers, depends on the evaluation level.

6.F.12 Two types of assumption about the environment are possible:

a) One subset could describe necessary external non-IT services on which the secure operation of the component relies (correct operation encompasses correctness and effectiveness in terms of ITSEC).

b) The other subset could describe necessary external IT services on which the secure operation of the component relies. For these IT services, there needs to be an interface description together with a description of services expected. This interface can be called the consumer interface of the component.

6.F.13 Any arbitrary combinations of assumptions are possible. For instance, a database management system may make purely IT-related assumptions regarding the provision of services by an underlying operating system. Whereas the underlying operating system may make non-IT assumptions about the physical security environment in which it operates.

6.F.14 The internal details are described to a granularity depending on the evaluation level. The internal details are a source of vulnerabilities.

6.F.15 A pictorial representation of a component is given in figure 6.F.1.

6.F.16 Using the above definition of the model of a component, a combination of components can be described. For simplicity, in the following presentation it is assumed that the set of assumptions about non-IT services is empty.

Combination of Components - Case 1

6.F.17 Component C1 uses services produced by component C2. An externally visible interface is delivered by component C1. C2 has a producer interface to component C1 but this interface is not visible to a user.

6.F.18 Example for Case 1:

a) component C1 - client;

b) component C2 - server.

6.F.19 A pictorial representation of Case 1 is given in figure 6.F.2. The arrow between the two components pointing at component C2 should be interpreted as C2 being used by C1.

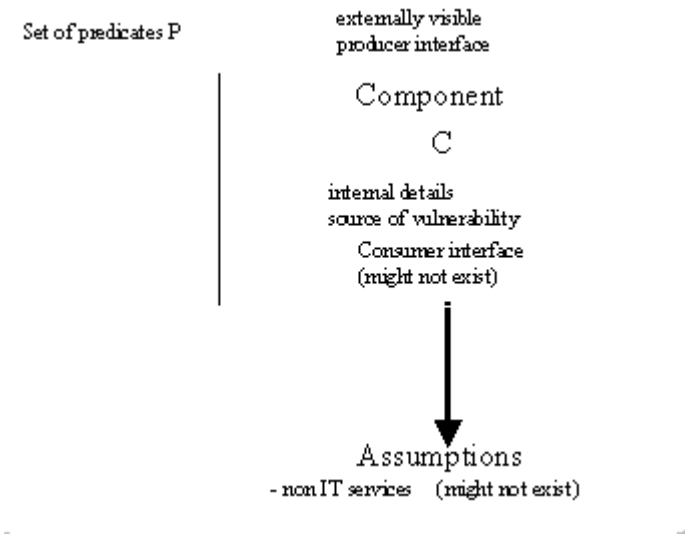


Figure 6.F.1 A TOE Component

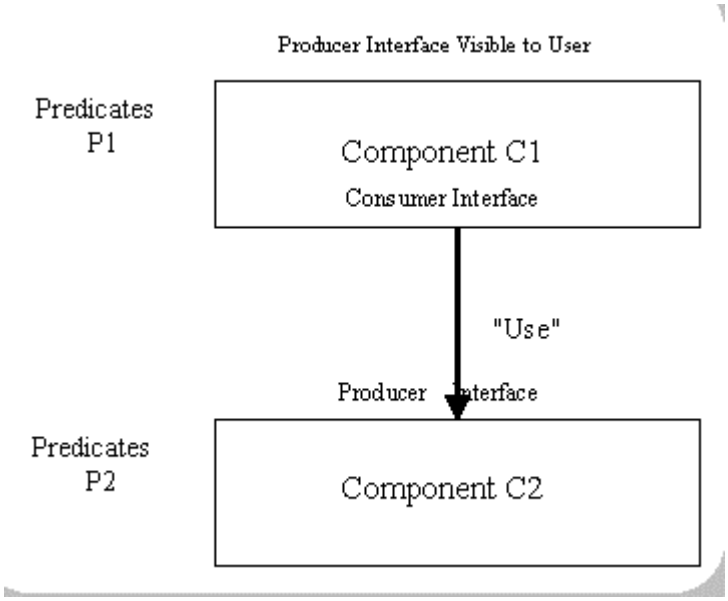


Figure 6.F.2 Combination of Components; Case 1

Combination of Components - Case 2

6.F.20 Component C1 uses services produced by component C2. Externally visible interface delivered by components C1 and C2.

6.F.21 Example for Case 2:

a) component C1 - virtual machine monitor (VMM);

b) component C2 - hardware platform.

6.F.22 The visible interface is provided by the VMM interface and Hardware platform's machine instructions.

6.F.23 A pictorial representation of Case 2 is given in figure 6.F.3.

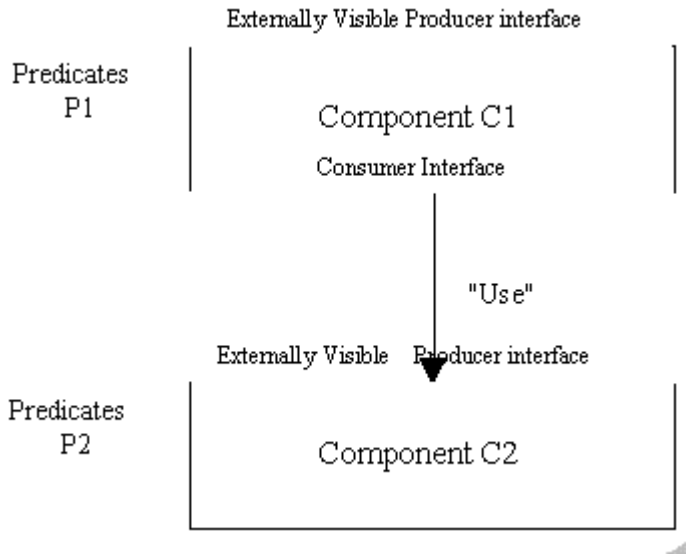


Figure 6.F.3 Combination of Components; Case 2

Combination of Components - Case 3

6.F.24 Component C1 uses services produced by component C2 and component C2 uses services produced by C1. External visible interface delivered by component C1 and component C2.

6.F.25 A pictorial representation of Case 3 is given in figure 6.F.4.

Compositions Resulting from Application of the Model

6.F.26 In all cases of combination the resulting component is called C3. It has its own set of predicates P3 and all the other features of a component such as producer interface, internal details etc.

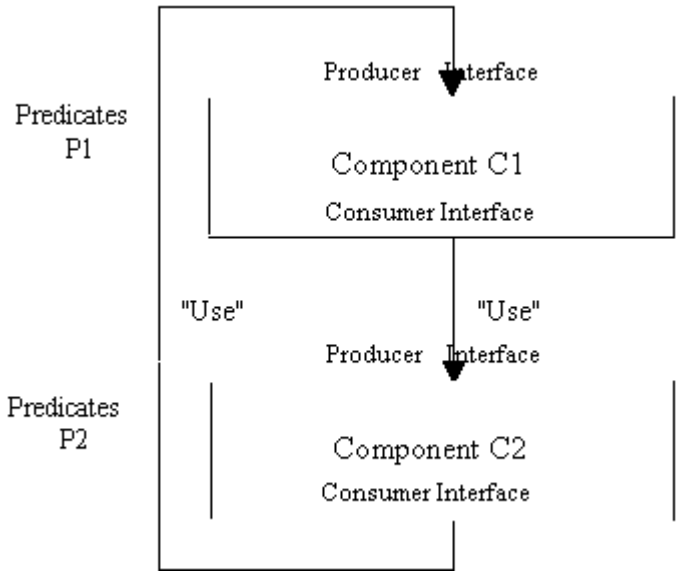


Figure 6.F.4 Combination of Components; Case 3

6.F.27 For example, if the combination is according to case 1, then for component C3 to maintain its predicates P3 the following conditions must be true:

Condition 1: C1 must be correctly implemented.

Condition 2: C2 must be correctly implemented.

Condition 3: The consumer interface of C1 must exactly match the producer interface of C2.

Condition 4: The combination of the predicates P1 and P2 results in the predicates P3. This means that P3 is derived from P1 and P2.

Condition 5: The predicates P2 must be preserved despite any vulnerabilities in C2. This means that the vulnerabilities in C2 must be shown not to be exploitable with respect to predicates P2.

Condition 6: The predicates P1 must be preserved despite any vulnerabilities in C1. This means that the vulnerabilities in C1 must be shown not to be exploitable with respect to predicates P1.

Condition 7: Vulnerabilities in C2 must be shown not to be exploitable with respect to predicates P1.

Condition 8: The use relation is really only from component C1 to component C2 (unidirectional).

6.F.28 The following is a list of possible problems in this scenario:

a) The confidence in the truth of condition 1 is different from the confidence in the truth of condition 2. This is the case where component C1 is evaluated to a different evaluation level than component C2.

b) Condition 3 is not true. Possible causes are:

- the consumer interface of C1 is a subset of the producer interface of C2;
- the consumer interface of C1 is a superset of the producer interface of C2;
- the interface descriptions of consumer (C1) and producer (C2) interface are at different levels of detail.

c) How to prove that predicates P1 and P2 make up P3 when P1 and P2 may be any set of predicates.

d) The confidence in the truth of condition 5 is different from the confidence in the truth of condition 6. This is again the case where component C1 and C2 are evaluated at different levels.

e) It has to be proven at the evaluation level specified that the use relation is as intended unidirectional.

6.F.29 If the representations of entities are of different granularity, which is normally the case if they reside at different levels of abstraction, then one can find a transformation so that the predicates of component C1 and component C2 at least reference the same entities. The created predicates P1' can now be investigated if they make up together with P2 the predicates P3 of the combination. It is essential to identify any contradictions of the two sets of predicates.

6.F.30 Due to the arbitrary nature of the predicates P1 and P2 no general rules can be given as to if and how predicates P1 and P2 make up predicates P3.

This page is intentionally left blank

Index

Terms from the ITSEC glossary are merked thus: ○