IT - Evaluation Manual

Manual
for the Evaluation of Trustworthiness
of Information Technology (IT) Systems

Version 1 of February 22, 1990

published by

GISA - German Information Security Agency
(ZSI - Zentralstelle für Sicherheit in der Informationstechnik)

on behalf of the Government of the Federal Republic of Germany

Bundesanzeiger

Foreword

The German Information Security Agency (GISA) drew up with representatives from academia and commerce and industry, the "Information Technology Security Criteria" on behalf of the Government of the Federal Republic of Germany and published them on 1st June 1989 as a means for the assessment of the security of IT systems.

This "IT Evaluation Manual" builds on the "IT Security Criteria". It was similarly drawn with the support and involvement of commerce and industry and academia and describes how IT systems or independent components are evaluated according to these criteria. It is intended to guarantee the equal treatment of both manufactures and their products which are to be evaluated.

GISA will only then issue a certificate when its tasks have been specified in due accordance with the law. In the transitional period it will restrict itself to evaluating such products for which a federal authority has notified a particular need. Nevertheless, GISA will develop criteria, procedures, tools and formal measures for the evaluation and assessment of the security of IT systems/components.

Apart from the granting of security certificates by GISA still to be regulated by law, a number of other questions in the "IT Evaluation Manual" have currently not been finally settled. At present there are no specific approved formal measures which correspond to all the requirements specified in Chapter 3 of this manual. First development results however make it certain that these measures will exist in the near future.

Of particular importance is the requirement to map the "IT security criteria" to the criteria catalogs of other nations. GISA is vigorously pursing the necessary activities for the harmonisation of security criteria and the mutual recognition of evaluations.

The "IT Evaluation Manual" has, therefore, in parts only temporary validity in this first version. As with the "IT Security Criteria", it will be updated as required in order to include new knowledge and practical experience obtained from evaluations.

In addition to the "IT Security Criteria" and the "IT Evaluation Manual", an "IT Security Manual" (Manual for Secure Application of Information Technology) will be drawn up. These manuals form the "Standard Works on IT Security". They provide comprehensive information on the determination of the security requirements and enable the planning and realisation of the security measures resulting from there to be carried out.

Dr. Leiberich

Director of the German Information Security Agency

Summary

This evaluation manual is Volume Two of the three volume publication for IT-Security in the Federal Republic of Germany. It contains a multitude of statements and references dealing with organisational matters surrounding an evaluation.

The evaluation manual consists of 12 chapters. The first 4 chapters are concerned with IT security criteria <1>, the remaining chapters contain independent statements on the environment and conduct of an evaluation.

Chapter 1 contains a number of detailed examples which give an introduction as to how mechanisms proposed for individual basic security functions are rated. This rating process will be conducted for each mechanism of a basic security function.

Chapter 2 gives explanations on the individual classes of functionality and specifically points out that the number of classes of functionality is not limited to those described in the IT security criteria catalog.

Chapter 3 provides detailed explanations on the individual points of each assurance level. This chapter shall always be read in conjunction with the chapter on assurances in the IT security criteria catalog.

Chapter 4 contains details on the contents and scope of the documentation to be presented at an evaluation using the assurance levels Q1-Q3 as an example.

Chapter 5 explains the assurance aspect "quality of the separation from components not to be evaluated". What is described is why the separation mechanisms have to be included in the evaluation. This is illustrated clearly by means of separation mechanism examples.

Chapter 6 describes the evaluation environment. After the passing of the law for the establishment and responsibilities of GISA legal and organisational matters will be explained here.

The following chapters, Chapters 7-12, refer to the evaluation process.

<1> IT Security Criteria: Criteria for the Evaluation of Trustworthiness of Information Technology (IT) Systems.
Published by GISA - German Information Security Agency on behalf of the Government of the federal Republic of Germany

Chapter 7 describes the conduct of an evaluation. The individual subsections describe a possible organisational structure of an evaluation team. Information on the review process is given next. This is the path suggested for reaching acceptance for individual and overall decisions during the evaluation. The next subsection reveals the preliminary work necessary if either a manufacturer or a user wishes to initiate an evaluation with the evaluation authority. A detailed suggestion in five phases with several steps follows which describes the sequence of an evaluation and the tasks to be tackled. Thereafter follows a possible procedure for a partial evaluation. At the end of the evaluation, a certificate is issued, the content and structure of which are described. The chapter closes with indications as to the consequences for a manufacturer when he markets an evaluated product.

Chapter 8 describes the particularities of developmental evaluations.

Chapter 9 explains when a re-evaluation becomes necessary. Four rules are stated and their application and the resulting consequences for the certificate are shown.

Chapter 10 deals with the evaluation of IT systems which already contain evaluated components. Here the knowledge from the first evaluation of such a system will still have a strong influence on the individual formulations. However, it is certain that putting together equally rated individual components does not necessarily produce the same assurance level.

Chapter 11 describes the structure of a tool and methods list which is absolutely essential for the manufacturer of IT systems of higher assurance levels. It is understandable that the evaluation authority can only support a small number of methods and tools which must be well-known.

Chapter 12 contains the mapping to other criteria catalogs. This is represented by the example of the mapping to the classes of the "Trusted Computer System Evaluation Criteria" of the American Department of Defense.

A glossary of the most important terms used in the IT evaluation manual then follows.

This edition (first version published on 22.02.1990) of the IT evaluation manual will need to be updated continuously as a result of the continuous developments in the IT field. The issuing authority is therefore open to constructive criticism to improve this volume.

Table of Contents

Foreword i

Summary iii

Table of Contents 1

1. Assessment of Mechanisms 3

Examples For The Rating Of Mechanisms 5
Basic Security Function Identification And Authentication 5
Basic Security Function Administration Of Rights 10
Basic Security Function Verification Of Rights 14
Basic Security Function Audit 16
Basic Security Function Object Reuse 17
Basic Security Function Error Recovery 19

2. Explanations of the Classes of Functionality 23

3. Explanations of the Assurance Criteria 26

Assurance Level Q1 27
Assurance Level Q2 30
Assurance Level Q3 34
Assurance Level Q4 38
Assurance Level Q5 42
Assurance Level Q6 47
Assurance Level Q7 51

4. Explanations of the Documents Required 54

4.1 Description of the Security Requirements 55

4.2 Specification of the System Components to be Evaluated 56

4.3 Description of the Separation of the System Components not to be Evaluated and the Interfaces to These Components 61

4.4 Documentation for the User 61

4.5 Description of the Hardware and Firmware Used with the Presentation of the Functionality of the Protection Mechanisms Realised in the Hardware or Firmware 62

4.6 Test Documentation 63

5. Explanations on Separation 64

6. Evaluation Environment 67

7. Description of the Evaluation Process 68
General Remarks 68

7.1 Organizational Structure of the Evaluation Team 68

Project Manager 68
Technical Project Leader 69
Evaluators 69
Moderator 70

7.2 The Review Process 70

7.3 How to Start an Evaluation 72

7.4 Sequence of an Evaluation 74

Phase 1 Establishing Contact 74
Phase 2 Examination Of Documents 75
Phase 3 Evaluation Of The Content Of Documents And Object 77
Phase 4 Preliminary Work For Issuing The Certificate 78
Phase 5 Preparation Of The Certificate 78

7.5 Assessment Steps of an Evaluation 80

7.6 The Certificate 81

7.7 Consequence for the Manufacturer 83

8. Developmental Evaluation 84

9. Reevaluation 86

10. Evaluation of IT Systems which Contain Already Evaluated **Components** 91

11. Tools and Methods 93

12. Mapping on Other Catalogs of Criteria 94

Glossary 97

1. Assessment of Mechanisms

Security requirements made on an IT system are enforced by security functions. Mechanisms are those methods and procedures by means of which these security functions are realised in a system. The means of rating the mechanisms is an essential part of the evaluation of a system. Thus the aim is to investigate whether the mechanisms applied are capable of enforcing the necessary security requirements adequately. Chapter 4 of the IT security criteria catalog describes possible vulnerabilities for each basic security function mechanism and is the starting point for the rating. In the evaluation what is sought are vulnerabilities by including all recognizable details of the realisation right down to the most detailed specification level together with the help of specific tests. Vulnerabilities exist when the effectiveness of a mechanism is limited, taking its embedment in the system into account either generally or in particular situations.

If individual details which are necessary for the mechanism rating cannot be derived from the specification, they have to be clarified by tests. The task of the tests is not to look for implementation errors but they are intended to help clarify such mechanism details which are not identifiable from the specification.

If vulnerabilities are found, a down-rating of the mechanism can take place. Whether a down-rating has to take place depends on the impact these vulnerabilities have on the enforcement of the security policy of the whole system.

A mechanism does not have to be down-rated if one (or more) of its vulnerabilities is (are) compensated by other mechanisms in such a way that the security policy is enforced in all conceivable situations by the combination of these mechanisms. Therefore it is entirely possible that one and the same mechanism is rated differently in two different systems with differing security policies.

The investigation of the individual mechanisms can still be undertaken independently of other mechanism applied and also independently of the concrete security policy. However, before a final rating of the mechanism in the respective system, the impact of the vulnerability on the conformance of the security policy has to be analysed whereby possible compensations by other mechanisms have to be taken into consideration.

The rating of a mechanism where vulnerabilities have been discovered then depends on what knowledge and what effort are required to exploit these vulnerabilities and how the threat as a result of the exploitation of the vulnerabilities is to be assessed. As this threat is in general not objectively measurable and also depends on the individual operational environment, the evaluation team has to judge the effect relatively subjectively in each individual case.

The interaction of several vulnerabilities can lead to a more severe down-rating than appears necessary when judging the individual rating of the vulnerability. Under no circumstances can a mechanism be rated higher than the lowest individual rating.

The rating results of earlier evaluations cannot be transferred untested in later evaluations because an evaluation can be altered by new technological developments. Thus, for example, the exploitation of a vulnerability can today make relatively high technological demands which, in future, will require considerably less effort.

Several mechanisms will be roughly outlined in the following examples and their vulnerabilities identified. What is not explained is how these vulnerabilities were found but rather it is assumed that these vulnerabilities were either identified from the system specification or were found by testing. The purpose of the examples is to explain the rating procedure whereby most of the examples selected are based on mechanisms from real systems but usually they have been simplified considerably. Only very few assumptions concerning the security policy were made. Therefore the results of these example ratings cannot usually be used directly in an evaluation.

Examples For The Rating Of Mechanisms

Basic Security Function Identification And Authentication

Example: Identification and Authentication of Users by a Combination of User-Id and Password

Inherent Vulnerabilities of the Mechanism

Passwords can also be passed on outside the system.

Passwords are frequently selected in such a way that they can be guessed easily.

Rating due to these vulnerabilities:

The combination of user-id and password can at the most be rated as a "very strong" mechanism due to these vulnerabilities.

Rating of the uniqueness of an identity:

What must be evaluated here is whether uniqueness is enforced by the IT system, i.e. whether a user-id can only be granted once.

If not, then the down-rating is dependent on the number of possible user-ids and the necessary organizational measures as well as their documentation in the appropriate manuals. This is an example of a vulnerability for which only the operating authority can rate the impact on its particular system. The rating depends on the number of users in his system, i.e. the operating authority itself must decide whether this number is so low that the organizational measures for enforcing the uniqueness for its system are sufficient. In this case the objective of the evaluation can only be to point out this vulnerability.

For the three threats listed in the criteria in the event of authentication by something a user knows, what must be examined for the mechanism of identification and authentication of users by a combination of user-id and password are:

- whether and with what effort a password can be "spied out" during input,
- whether and with what effort a so-called "spoofing" program can be implemented, which can cheat the user into involuntarily revealing his password,

- how and where passwords are stored in the system, how access to these areas is governed and how much effort is necessary for a "Trojan horse" to access passwords and pass them on to unauthorized persons.

In accordance with the relevance of these threats, the mechanism for the identification and authentication of users by a combination of user-id and password shall be evaluated.

The following section is intended to illustrate what the rating of the mechanism for identification and authentication of users could look like in the evaluation report.

Description of the Mechanism

A combination of user-id and password is used for the identification and authentication of users. Passwords are a maximum of 8 characters long, whereby each character can originate from a broad range of the ASCII alphabet. The possible range of passwords is therefore very large (more than 10^{17} possibilities). Passwords are only stored internally in encrypted form. A one-way encryption is used for this. The strength of the encryption algorithm has not been examined.

Security Requirements to be Enforced by the Mechanism

Users shall be unambiguously identified and authenticated by the mechanism. The authentication data shall be protected against unauthorized access of any kind.

Rating of the Inherent Vulnerabilities of the Mechanism

Due to the inherent vulnerabilities of the mechanism (conscious or unconscious passing on of passwords, inadequate selection of passwords), the mechanism can at the most be rated "very strong".

Rating on the Basis of Criteria

1. The enforcement of the uniqueness of an identity is not given.

   ===> light down-rating

   Justification:
   The possible impact of this threat depends very strongly on the operational environment of the system and the number of users.

2. The password is not displayed on the screen when entered and (normally) is not stored in the local memory of the display. However what can occur is that an unconcentrated user enters his password before he is requested to do so. In this case the text entered on the screen is then displayed. However this is only considered to be a minimal threat.

   ===> no down-rating

   Justification:
   The threat can be reduced by instructing the user. It is not possible to exploit this vulnerability selectively.

3. A "spoofing" program is very easy to implement and is hardly noticed by the user. No special knowledge of the system is required to implement such a program.

   ===> strong down-rating, due to this vulnerability the mechanism can at the most be rated "moderate".

   Justification:
   A malicious user does not require any special knowledge or support to implement such a spoofing program. Well-meaning users can hardly protect themselves against a threat evolving from this vulnerability. Thus according to the criteria for the rating of mechanisms, the maximum rating possible here is "moderate".

4. Passwords are in fact stored in encrypted form in a file which is readable by all users. The encryption algorithm is known. This constitutes a vulnerability. The range of possible passwords is very wide (more than 1017 possible passwords). Despite this, the protection of the passwords in this system is to be considered inadequate.

   ===> down-rating, due to this vulnerability the mechanism can at the most be rated "strong".

   Justification:
   With this kind of password protection the following attack is possible: Passwords are selected, encrypted and tested as to whether the encrypted text occurs in the password file. For this attack knowledge of the system (of the encryption algorithm) is required and a relatively large computing effort is required to check through a relatively large number of candidates to achieve success. A well-meaning user can protect himself relatively well against a threat through this vulnerability by selecting an appropriate password.

Rating of the Mechanism

Altogether the mechanism used for the identification and authentication of users can only be rated as **"moderate"**, since it can be overcome with medium effort by persons with normal knowledge of the system.

Impact on the Overall Rating of the System:

Identification and authentication of users is an important aspect of the security policy of the system. Due to the rating of the mechanism for identification and authentication of users, the system can **at the most achieve assurance level Q2**.

Example: Identification and authentication using a machine-readable id-card and a personal identification number (PIN) which is not freely selectable. The id-card can only be forged with great effort.

Inherent Vulnerabilities of the Mechanism:

Codes can also be passed on outside the system.

Id-cards can fall into the hands of unauthorized persons outside the system.

Both conditions must occur simultaneously in order to make the threat of wrong identification and authentication effective. In view of this vulnerability the mechanism cannot be rated higher than "very strong".

Description of the Mechanism

Each id-card bears a clear identification mark which is machine-readable. The PIN consists of 4 decimal digits. This provides a volume of passwords of 10000 possible codes. The number of permitted identification attempts is restricted to 3. The code is pre-allocated, thus it is not freely selectable by the user.

Security Requirements to be Enforced by the Mechanism

The mechanism is intended to identify and authenticate users. Authentication information is to be protected against knowledge by unauthorized persons. The id-card is to be protected against forgery and unauthorized manipulations.

Rating on the Basis of the Criteria

Each id-card is unambiguously identifiable. Hence uniqueness is given. An expertise confirms that the production of a forged id-card is only possible with very substantial effort. (**Note:** The evaluation authority is unable to check this, however it can require the presentation of an expertise from an independent body in which these questions are clarified.)

Since the volume of possible codes is restricted to 10000, the probability of wrong identification by trial and error is roughly 3/10000. The id-card can also easily be stolen. Hence due to this vulnerability, the mechanism can at the most be rated "moderate".

Examinations have revealed that by relatively simple manipulations on the id-card, the number of allowable false attempts at identification and authentication can in effect be increased at will. This leads to a further down-rating.

Rating of the Mechanism

On the grounds of the vulnerabilities found the mechanism can only be evaluated as **"weak"**.

Impacts on the Overall Rating of the System

Due to this rating the system can **at the most achieve assurance level Q1**, since the identification and authentication of users is considered to be an important component of the security policy.

Basic Security Function Administration Of Rights

Example: Administration of Rights with Access Control Lists

Description of the Mechanism

Access control lists which are kept in special files are implemented for the administration of rights. These files are protected against unauthorized access. System functions exist for granting and altering rights. Rights to an object can only be granted by the "owner" (creator) of the object and the system administrator. The right most recently issued is the valid one. Rights can only be established explicitly and between individual subjects and individual objects. The access control lists are not updated when a user is deleted or renamed. Users authorized to access an object can lock this object and therefore deny access to other authorized users.

Security Requirements to be Enforced by the Mechanism

The mechanism is to administer access rights to files of users. It should be possible to specify the access rights separately for each user and for each file. Only the owner of the file and the system administrator should be allowed to grant, revoke or alter access rights.

Rating on the Basis of the Criteria

Completeness
Completeness is given. It is possible to establish individual access rights for each user and each file.

Uniqueness
Ambiguity can only occur when the system administrator and the owner grant differing rights relating to an object. This conflict is resolved by the fact that the right most recently granted is the valid right. If both the system administrator and the owner are in a position to have the access rights to an object listed, then uniqueness is guaranteed.

Clarity
Both the system administrator and the owner of an object can display all the access rights granted to this object. Since no implicit rights exist (e.g. established by a set of rules), clarity is given. However it is not possible for a user to list for himself the objects to which he possesses access rights and the nature of these rights. This is only considered to be a very slight vulnerability, which does not involve any down-rating.

Protection against covert alteration of rights
The system functions for the administration of rights can be called from normal user programs. In this way rights can be established, deleted or altered in a covert manner (e.g. by means of "Trojan horses"). It is also possible for the system administrator to block himself, i.e. to revoke his own rights.

===> down-rating, the mechanism can at the most be rated "strong".

Justification:
This vulnerability represents a threat which must be taken seriously. It permits rights to be established, altered or deleted by unauthorized persons. However a user authorized to do so must explicitly start the program which performs the covert alteration of rights.

Administration of rights in connection with the deletion or renaming of subjects or objects
The access control lists are not updated when a subject is deleted or renamed. This can lead to undesired right relationships. This vulnerability represents a moderate threat.

===> down-rating, on the grounds of this vulnerability the mechanism can at the most be rated "very strong".

Protection against restrictions in the ability to exercise rights
The system allows objects to be locked by users with access authorization. No measures have been taken to restrict the duration of this lock. The security policy contains no availability aspects. For this reason this vulnerability does not involve any down-rating, but it is pointed out that this system should not be used in environments involving high demands as to the availability of data from files.

Rating of the Mechanism

Due to the vulnerabilities shown the mechanism is rated **"strong"**.

Impact on the Overall Rating of the System

As a result of this rating the overall system can **at the most achieve assurance level Q4**.

Example: Capabilities

Description of the Mechanism

A distributed system in which access rights are administered via capabilities is examined here. The capabilities are protected against unauthorized modifications by an encryption mechanism. The encryption mechanism used has been rated "virtually unbreakable". The revocation of capabilities is realized by modifying the key. However as a result of this all the capabilities previously granted to this object then loose their validity. A garbage collection for objects for which no capabilities exist any longer is available as a privileged procedure which can be started by users with the appropriate capability. The passing on or copying of capabilities is possible from every program.

Security Requirements to be Enforced by the Mechanism

The mechanism should make it possible to administer access rights between users and processes to files, programs and processes. It shall be possible to pass on in a controlled manner a right possessed by a user or a process. It shall be possible to revoke rights.

General Remarks

Known problem areas exist using this mechanism to which special attention shall be paid in the examination and rating. These problem areas are:

- revoking of capabilities
- restriction of the passing on of capabilities
- existence of objects for which no more capabilities exist in the system.

Rating on the Basis of the Criteria

Completeness
The granularity of the rights conforms with the security policy. Hence the completeness aspect exists.

Uniqueness
Since the withdrawal of capabilities for an object is only possible on a global basis, uniqueness is guaranteed.

Clarity
A user can list his capabilities and thus knows what access rights to which objects he has. He thus has all the necessary information. It is not necessary for him to know the names of objects to which he possesses no access rights. Normally it is not necessary to create a list of subjects with authorized access to an object. Such a function is only expedient and exists in order to check whether any subject at all in the system still possesses a capability for this object ("garbage collection").

Protection against covert alterations
Since capabilities can be passed on or copied from every program, no protection is given against covert passing of rights. Thus rights can be passed on at will, for instance by means of "Trojan horses".

===> down-rating, the mechanism can at the most be rated "strong".

Justification:
This mechanism allows the unintentional passing on of rights which subsequently are very difficult to revoke. A well-meaning user can only partially protect himself against this threat.

Administration of rights for deleting or renaming subjects or objects
When a subject is deleted, the capabilities are deleted with it; when a subject is renamed, the capabilities remain unchanged. When an object is deleted the capabilities in connection with this object are not deleted automatically, the user simply learns that this object no longer exists. When an object is renamed the capabilities retain their validity.

Protection against restriction of the ability to exercise rights
The system allows objects to be locked by users with authorized access. No measures have been taken to restrict the duration of this lock.
The security policy contains no availability aspects. For this reason this vulnerability entails no down-rating, but it is pointed out that this system should not be used in environments involving high demands as to the availability of data from normal files.

Rating of the Mechanism

Due to the vulnerabilities shown the mechanism is rated **"strong"**.

Impacts on the Overall Rating of the System

As a result of this rating the overall system can **at the most achieve assurance level Q4**.

Basic Security Function Verification Of Rights

Example: Evaluation of the Rights When Setting Up a Logical Connection

Description of the Mechanism

A system is assumed which administers access rights of users to files using access control lists. The check for authorized access is performed at open time. No subsequent check is performed at the actual access. If an access right is revoked it is not verified whether the user whose right has been revoked still has the file open.

Security Policy to be Enforced by the Mechanism

Each time a user tries to access a file the mechanism should verify the validity of this access.

Rating on the Basis of the Criteria

Completeness of the verification of rights
The specification does not reveal any channels by which access to data in a file is possible without first opening this file. (**Note:** Due to implementation errors such channels can naturally exist in real systems. Such errors are sought during the quality testing of the system, but not during the evaluation of the mechanism.)

Time of the verification of rights
Verification is performed prior to actual access when a file is opened. The actual accesses may take place much later. If the access right is revoked in the meantime, the user can still access the file as long as it is still open. It is left to the user's discretion how long he keeps the file open.

===> down-rating, the mechanism can at the most be rated "strong".

Justification:
This vulnerability can be exploited by malicious users. However the condition for this is that they have previously owned the access right. For this reason the threat created by the exploitation of this vulnerability is not considered to be too serious. For this reason it is possible to rate this mechanism as "strong", despite the vulnerability.

Availability of decision-making data
Due to hardware malfunctions (e.g. failure of the disk where the access control lists are stored) a situation can develop in which the decision making data is no longer available but the system would in principle be in a position to partially fulfil its services. However in this situation the system denies all users access to files. As a result of this none of the services of the system are available.

Whether and to what extent the mechanism has to be down-rated as a result of this vulnerability depends on the degree of probability with which the situation described above can occur.

Integrity of the decision making data
The access control lists are stored in a file. Access to this file is monitored via the mechanisms of administration of rights and verification of rights. Selective manipulation of the access control lists by unauthorized persons is therefore largely excluded by the design. Check sums are used to identify errors while stored. Hence the probability of unidentified stochastic errors in the data stored is so low that no down-rating is necessary.

Rating of the Mechanism

Due to the vulnerabilities noted the mechanism is rated **"strong"**.

Impact on the Overall Rating of the System

As a result of this rating the overall system can **at the most achieve assurance level Q4**.

Basic Security Function Audit

Example: Auditing Using Ordinary Files Accessed by System Functions

Description of the Mechanism

Auditing is performed on ordinary files which can be protected by the mechanisms of administration of rights and verification of rights. There is no automatic protection. Auditing is called up via the system interfaces.

Security Requirements to be Enforced by the Mechanism

The mechanism shall audit each use of the identification and authentication mechanism and each attempted access to files with date, time and user name. The mechanism shall protect audit data against unauthorized access. The compiled audit data shall be considered as evidence.

Rating on the Basis of the Criteria

Non-deceivability of the audit
Tests proved that the user programs can generate audit records with any content desired. It is thus possible to audit "events" which have not in fact happened. Thus the mechanism can be deceived.

===> very strong down-rating, the mechanism can only be rated "ineffective".

Justification:
The mechanism is not in a position to enforce the demands made of it since the audited information cannot be considered as evidence.

Completeness
All the events listed in the security requirements are audited with the required information.

Rating of the Mechanism

On the grounds of the vulnerability indicated the mechanism is rated **"ineffective"**.

Impact on the Overall Rating of the System

As a result of this rating the overall system can **only achieve assurance level Q0**.

Basic Security Function Object Reuse

Example: Overwriting the Contents of Deleted Files with Binary Zeros

Description of the Mechanism

When a file is deleted, not only is the catalog entry deleted but also the space on the storage media allocated for this file is overwritten with binary zeros.

Security Requirements to be Enforced by the Mechanism

The space on the storage medium allocated for a file shall be prepared for reuse when this file is deleted in such a manner that it is subsequently no longer possible to infer any information previously stored in this file.

Rating on the Basis of the Criteria

Nature of the object reuse
In the case of removable storage media there is a risk that they may be analyzed outside the system. With costly procedures it may under certain circumstances be possible to reconstruct the content of deleted files. Whether and with what effort this is possible is not examined within the context of the evaluation. It is not possible to reconstruct the content of deleted files using system functions.

===> slight down-rating, the mechanism can at the most be rated "very strong".

Justification:
There is a possibility that the data might be able to be reconstructed on the storage media by means of an extremely costly analysis. This risk can be countered by organizational measures. The enforcement of these measures cannot be monitored by the IT system, however. For this reason according to the rules of the catalog of criteria the mechanism can at the most be rated "very strong".

Time of object reuse
The object reuse is a part of the system function for deleting a file. After termination of the deletion process it is no longer possible to access the data of the deleted file.

Rating of the Mechanism

On the grounds of the vulnerability indicated the mechanism is rated **"very strong"**.

Impact on the Overall Rating of the System

As a result of this rating the overall system can **at the most achieve assurance level Q6**.

Basic Security Function Error Recovery

Example: Handling of Program Errors by the Operating System

Description of the Mechanism

Certain program errors identified by the hardware or the firmware (e.g. access attempt outside the main memory area available, invalid machine instructions) generate an interrupt. When such an interrupt occurs the operating system analyzes which program generated the interrupt and then terminates this program.

Security Requirements to be Enforced by the Mechanism

In the event of the program errors listed below the system should terminate the program which caused the error in a controlled manner. All write operations on files or other external storage media initiated by the program up to the time when the error occurred must be completed in a controlled manner. No data may be lost in this process.

The following errors must be identified and treated:

- Access to protected main memory areas .
- Attempt to perform invalid or privileged operations.
- Division of an integer number by zero.

Rating on the Basis of the Criteria

Completeness of error identification
It is apparent from the design documents of the processor that there are no exceptions in the treatment of errors. Nor were any exceptions found during the tests.

Correctness of the error analysis
For this mechanism the following information is required for analysis of the error:

- The nature of the error.
- The address of the instruction which caused the error.
- The process or the program which caused the error.

It is apparent from the design documents of the processor that the nature of the error can be determined unambiguously by an interrupt. It is also apparent from these documents that the instruction which caused the error is unambiguously identifiable. It is clear from the system specification that it can be determined at any time which process or which program is active.

Tests did not reveal any traces of wrong error analysis.

Loss of data, functionality or timeliness
In the event of errors programs are terminated by the operating system. Thus we have a loss of functionality. In the event of an error, output buffer stored in main memory are not flushed. In the event of an error, files are not closed properly. This can result in a considerable loss of data. Loss of timeliness results from the necessity for reconstructing data and restarting the program which caused the error.

===> very strong down-rating, the mechanism can only be rated as "ineffective".

Justification:
The security policy requires that even in the event of an error all data written up to this point in time are available. The mechanism is not in a position to satisfy the security policy it is intended to enforce. It must therefore be rated "ineffective".

Independence of the error recovery from the source of error
No way was found to influence the actual error recovery as a result of the occurrence of the error.

Error in the error recovery
It is quite possible that one of the errors to be recovered from may occur in the error recovery itself which is performed by the operating system. No precaution have been taken to identify such a case and to deal with it specifically. This can result in recursive calls to the error recovery program. However the probability of such a case occurring is very slight.

===> down-rating, as a result of this vulnerability the mechanism can at the most be rated "strong".

Justification:
The low degree of probability of the occurrence of such a case explains why the mechanism can still be rated "strong" despite this vulnerability.

Rating of the Mechanism

Due to the vulnerabilities found the mechanism can only be rated **"ineffective"**.

Impact on the Overall Rating of the System

As a result of this rating the overall system can **only achieve assurance level Q0.**

Example: Error Recovery in Remote Data Transmission by Error Identifying and Error Correcting Communication Protocols

Description of the Mechanism

With each data packet a check sum is transmitted. This check sum is designed in such a way that any errors in an 8-bit long portion of a data packet can be identified reliably. The protocol allows for the receiving end to notify the sender upon reception of each data packet whether the data has been received correctly. If the data is not received correctly, the data packet is retransmitted. If a data packet is not received correctly after the third attempt, the communication is aborted.

Security Requirements to be Enforced by the Mechanism

The mechanism is intended to identify and correct unintentionally caused errors in the data transmission.

Rating on the Basis of the Criteria

Completeness of error identification
The level of the unidentified errors and the degree of probability of the occurrence of such an error must be determined. The mechanism is then to be down-rated on the basis of this probability. As a basis for the down-rating of the mechanism it is generally more expedient to determine the probability of the occurrence of such an error per day taking as a basis the average data traffic carried the line. In this way the volume of data transmitted over the line is included in the rating. The following may apply as a rule of thumb:

Let P be the probability for the occurrence of an unidentified transmission error per day.

The following then applies:

If P is greater than 0.5, the mechanism is down-graded to "weak".

If P is greater than 0.1 and lower than or equal to 0.5, the mechanism is down-graded to "moderate".

If P is greater than 0.001 and lower than or equal to 0.1, the mechanism is down-graded to "strong".

If P is lower than or equal to 0.001, the mechanism is down-graded to "very strong".

If the demands relating to the data integrity are particularly weak or particularly high, then this rule of thumb must be adapted accordingly.

In the concrete example under review let P have been calculated to be 0.0005.

===> the mechanism can at the most be rated "very strong".

Correctness of the error analysis
No analysis of the source of error is conducted.

Loss of data, functionality or timeliness
A data loss only occurs in the event of an unidentified communication error. This case has already been rated above.
Loss of function only occurs when an identified error occurred in three consecutive transmission attempts. Here too the probability of the occurrence of such a case must be assessed. The rating due to this vulnerability depends on the requirements regarding the availability of data in the recipient system.
A small loss of time occurs with each recognized error due to the repeated transmission of the data packet. The probability of the occurrence of an identifiable error in a data packet must be assessed. Whether and to what extent the mechanism must be down-rated as a result of this vulnerability depends on this probability. In the concrete example the security policy does not mention any conditions regarding the availability of the data in the receiving system. If there were a requirement for immediate availability without loss of time , the mechanism would have to be down-rated correspondingly. However, in the specific example under review, no down-rating is necessary on the grounds of this vulnerability, but it must be pointed out in the evaluation report that this system is only partially suitable for applications which require immediate availability of the information transmitted to the receiving system.

Rating of the Mechanism

On the grounds of the vulnerabilities found the mechanism is rated **"very strong"**.

Impact on the Overall Rating of the System

On the grounds of this rating the system can **in general achieve the assurance level Q6** at the most. Only if the sponsor and the evaluation authority are in agreement that the data communication channels secured by this mechanism need not fulfil any particularly security relevant duties can classification at assurance level Q7 also be possible.

2. Explanations of the Classes of Functionality

Before selecting a system a user will demand that certain security functions exist which are derived from his threat analysis. If he wishes to select an appropriate product for his purposes from the evaluated product list, then it is expedient to give him guidelines as to which products fulfil his functional requirements and which products are not appropriate to his needs. That is the purpose of the classes of functionality.

The requirements in the individual classes of functionality are deliberately formulated very abstractly. A user should understand using this level of abstraction which classes of functionality he requires for his system. In general, his requirements are more detailed than the descriptions of the class(es) of functionality. The user can, however, restrict himself for the selection of a system to the consideration of those systems which were evaluated in the class(es) of functionality appropriate to his needs and at the assurance level for his requirements. For these systems he can then compare his security policy with the detailed description of the security functions in the individual evaluation reports. By the pre-selection via the classes of functionality, the user is spared the task of considering all the evaluated systems in his selection process.

It is, of course, also possible that a system, which was evaluated in the class of functionality required by the user, is nevertheless not appropriate to the needs of the user. This is always the case when for the global requirements of a class of functionality the user makes more detailed requirements that cannot be fulfilled any longer by all systems of this class of functionality. Thus, for instance, a user of the class of functionality F2 can require the formulation of the security policy to be so precise that he demands the access rights reading, writing and executing or a combination of these rights for files as objects of the administration of rights. A virtual machine monitor, which only knows the access rights reading, writing as well as reading and writing on the level of logical disks or disk segments, can in principle (if the other F2 requirements are also fulfilled) be evaluated in the F2 functionality class. However it is unsuitable for the user with the above-mentioned requirements.

On the other hand, this means that it is inadequate to specify only one or more classes of functionality with the assurance level aimed for when submitting a system for evaluation. A detailed list of the security requirements is absolutely necessary for the evaluation. These must cover the abstract requirements of the class(es) of functionality aimed for and be formulations and extensions of these requirements.

Extensions can be additional security requirements which cannot be derived from the classes of functionality aimed for. These additional security requirements are also examined in the evaluation and described in the evaluation report.

The classes of functionality F1 to F5 are derived from the functionality of the classes of the American "Trusted Computer System Evaluation Criteria" (the so-called "Orange Book"). What is intended to be guaranteed is that, on the one hand, systems which were evaluated in the USA, in accordance to the criteria of the "Orange Book", can also be categorized in the national IT security criteria catalog with reference to its functionality. On the other hand, in particular instances, the reverse mapping is only possible when a system was evaluated for functionality and assurance in accordance with the national IT security criteria in classes which comprise the criteria of an "Orange Book" class. What this mapping looks like in individual cases is described in Chapter 12 of this manual. According to the classes of the "Orange Book", the classes of functionality F1 to F5 are hierarchically structured, i.e. the lowest requirements are made in the functionality class F1 whilst the highest requirements are made of the security functions in functionality class F5. The remaining classes of functionality do not have an equivalent class in the "Orange Book" and are not hierarchically structured. The requirements made of the individual basic functions were summarized for all functionality classes. Thereby the requirements made of the individual classes of functionality are intended to be more comprehensible.

The requirements in the "Orange Book" were re-phrased in some places and partially generalized, as they did no fit the IT security criteria philosophy in the form presented there. This is particularly valid for the cases in which the "Orange Book" prescribes mechanisms (such as for "labels", for address spaces, etc).

The "mandatory access control" was also generalized, as the rules set out in the "Orange Book" are not meaningful for all system environments. What is required is that the rules formulated in the "Orange Book" can be realized (such as by means of a special configuration of the system).

For the classes of functionality, which do not have any equivalents in the "Orange Book", i.e. in the classes of functionality F6 to F10, no attempt is made any longer to cover as many basic security functions as possible. Basically, each of these functionality classes makes security demands of a special security function. Functionality class F6 alone is the exception here as strongly inter-dependent requirements of basic security functions such as "identification and authentication", "administration of rights", "verification of rights" and "audit" are made.

Thus it should be possible to define the security requirements for the individual basic security functions independently of one another to a large extent (in as far as this is meaningful) and to select classes of functionality, which cover the stated security requirements. It is therefore entirely feasible and also sensible that a system enforces the security requirements of several classes of functionality or security requirements are stated for the system which are not included in any of the previous classes of functionality in this combination. All these classes of functionality are listed in the certificate. Possible additional security functions which go beyond the requirements listed in the certificate made of the functionality classes are set out in the certificate and are described in the evaluation report in full.

It is possible to define new classes of functionality at any time and to include them in the IT security criteria as an addendum. Whether new classes of functionality are included and how these look like is left to the discretion of the evaluation authority. However proposals for such new classes of functionality should be put forward from the outside to the evaluation authority. If new classes of functionality are included in the IT security criteria, the sponsors of earlier evaluations can apply for a review, whether their already evaluated products enforces the criteria of a new class. For this, no new product review is necessary but the requirements of the new class are compared with the security function described of the evaluated system in the evaluation report. If it is clear that the system satisfies the requirements of the new class of functionality, this is confirmed in an addendum to the certificate. Only if by means of this comparison it can not be unambiguously decided whether the evaluated system fulfils the requirements of the new functionality class, a (in general very short) product reexamination is necessary by means of which the existing ambiguities are intended to be clarified.

As has already been explained, it is always possible to evaluate systems which do not fulfil all the criteria of one of the listed functionality classes in the IT security criteria catalog. In this case, the security functions of the system will be completely described in the evaluation report. Only the assurance level achieved is recorded in the certificate for such systems and reference made to the description of the security functions in the evaluation report.

3. Explanations of the Assurance Criteria

The following chapter contains explanations of the individual assurance levels. This is intended on the one hand to contribute towards a better understanding of the assurance criteria, and, on the other hand, to provide help for applying the criteria in the evaluation process. Although this chapter is designed completely analogously to the corresponding chapter in the IT security criteria, in contrast to the IT security criteria, it is important here to note also all the explanations provided for the lower assurance levels. Thus essentially for each assurance level only those aspects which are either quite new and required additionally or which have changed by comparison with the next level down are discussed. Difficulties in interpretation which will arise during the first evaluations should be eliminated by additional explanations in the IT Evaluation Manual. This will probably lead to the IT Evaluation Manual being modified to a greater extent than the actual IT security criteria.

The explanations of the individual assurance levels are given on the following pages. No explanations have been provided for assurance level Q0.

Assurance Level Q1

Explanation of the Criteria

Assurance level Q1 is intended for systems or system components which do not need to satisfy any particularly high requirements as regards the assurance of the enforcement of the security policy. The evaluation here is merely intended to ensure that the implementation more or less enforces the security policy and that no major errors exist. Systems of assurance level Q1 can certainly be adequate for areas which are not security-critical. They ensure that the security policy is enforced by well-meaning user behaviour, but despite the evaluation a relatively high residual risk remains that vulnerabilities still exist in the system through which the security policy can be violated.

Quality of the Security Policy

Explanation of the Criteria

In order to achieve assurance level Q1 the security policy need only be specified very roughly and superficially and can leave substantial leeway for interpretation. However no clearly identifiable ambiguities may be found upon first reading of the document describing the security policy. Any ambiguities found during the evaluation which may be based on certain interpretations of the security policy shall then be clarified between the sponsor and the evaluation authority.

Quality of the Specification

Explanation of the Criteria

It shall be possible to derive from the specification what mechanisms are used to enforce the security policy, even if not all the details of these mechanisms are described. However either the description shall be sufficient to rate the mechanisms, or it shall be possible to determine the missing information needed to allow rating by means of simple tests. However the effort required to perform these tests shall be so low that they can be implemented within the scope of the evaluation plan.

Quality of the Mechanisms Used

Explanation of the Criteria

The minimum rating for a mechanism, which is solely responsible for the enforcement of a particular security requirement, shall be "moderate". A mechanism rated as "weak" shall only be used when at least one of the following conditions is fulfilled:

- The mechanism is used in conjunction with other mechanisms. The combination of these mechanisms is rated "moderate" or higher.
- The mechanism serves the purpose of enforcing a security requirement which, in the opinion of the sponsor and the evaluation team, only plays a subordinate role.
- The effort to apply a stronger mechanism does not, in the opinion of the sponsor and the evaluation team, justify the costs involved in achieving the higher level.

Quality of the Separation from the System Components not to be Evaluated

Explanation of the Criteria

Separation of the security functions from the system components not to be evaluated is also essential for achieving assurance level Q1. Systems which do not provide this or only use a mechanism rated "moderate" or even "weak" for this purpose can only be classified in assurance level Q0. The tests for this area are essentially merely intended to show that the interfaces between the system components to be evaluated and those not to be evaluated basically behave as described in the documentation.

Quality of the Software Development Process

Since for an evaluation at assurance level Q1 the source code of the implementation need not be presented, it is not expedient to specify requirements with regard to the implementation language, the implementation environment or the internal structure of the source code. For this assurance level the evaluation of the implementation simply consists in performing a series of tests intended to demonstrate that under normal use the system enforces the security policy.

Specific sophisticated penetration tests are not necessary. However these simple tests may not reveal violations of the security policy. The selection of the tests and the rating of the vulnerabilities found is left to the evaluation authority.

Quality of the Operational Behaviour

Explanation of the Criteria

The sponsor specifies prior to the beginning of the evaluation which configurations (hardware and software) are to be evaluated. The requirements regarding the configuration state that someone who configures a system must be able, after reading the documentation, to assess the impact of the configuration he has selected on the essential points of the security functions of the system. A few of the configuration possibilities are to be selected for testing. The system should be configured in this way and the selected test cases should be run (whereby these test cases are essentially the same for all configurations and are only adapted when the test cannot be implemented or is not suitable in its original version for the configuration selected).

Quality of the User Oriented Documentation

Explanation of the Criteria

The user-oriented documentation shall be easy to handle and describe all the security-relevant functions to the user of the system both comprehensively and comprehensibly. In the event of divergences between the real system behaviour and the description in the user-oriented documentation, the sponsor must always be granted a timeframe to correct the documentation. The sponsor is to be notified of all ambiguities found.

Assurance Level Q2

Explanation of the Criteria

Assurance level Q2 is intended for systems or system components of which moderate requirements are made as regards assurance of enforcement of the security policy. The goal of the evaluation is to obtain a considerable degree of confidence that the security policy can neither be invalidated nor violated by easily exploitable faults. Systems of assurance level Q2 are frequently sufficient for areas with light to moderate requirements regarding the enforcement of the security policy. The evaluation in this class is intended to show that simple penetration tests revealed no errors that would allow the violation of the security policy.

Quality of the Security Policy

Explanation of the Criteria

The security policy for assurance level Q2 is generally only formulated in natural language and presents the objectives for the security functions. The relationship to the possible threats and to the basic security functions is to be presented in the security policy. A formal evaluation of consistency is not possible in this case. Hence only informal ambiguities can be sought. Should such an ambiguity be discovered, the point must be discussed with the sponsor, as it may be the result, under certain circumstances, of a misinterpretation of the security policy. In any case, the security policy has to be reformulated in such a way that ambiguities and uncertainties found do not reoccur. Thereby all informal ambiguities are eliminated from the security policy.

Quality of the Specification

Explanation of the Criteria

Although the specification may be a quite superficial description of the implementation formulated in natural language, there may be no uncertainties regarding the algorithms and mechanisms used. Moreover the specification shall relate to the security policy and explain clearly what part of the security policy is to be enforced with which algorithms and mechanisms. Only in this way is it possible for the evaluation authority to check the consistency between the security policy and the specification with a justifiable effort.

If uncertainties arise during this evaluation, they must be clarified with the sponsor or the manufacturer. In some cases, however, clarification can also be achieved by testing. Following clarification of all uncertainties the sponsor must present an appropriately updated version of the specification.

Side effects, by means of which the security policy may be invalidated or violated, are of course very difficult to find in an informal specification. However, uncertainties in understanding a specification often indicate the presence of such side effects. These are areas which have to be subjected to particularly careful tests.

In particular, such uncertainties include parameter values for the security functions the effects of which are either not described at all in the specification or only incompletely.

Quality of the Mechanisms Used

Explanation of the Criteria

The assessment of a mechanism as "moderate" states that it already offers reasonable protection against willful violations of the security policy. Thus such a mechanism may be regarded as fully adequate for the objectives of assurance level Q2.

Quality the Separation to the System Components not to be Evaluated

Explanation of the Criteria

The quality of the separation from the system components not to be evaluated is a very important aspect in the assessment of protection against penetration and manipulation of systems or individual components. Many system penetrations are based on vulnerabilities in this area. Such vulnerabilities are in particular:

- Inadequate evaluation of parameters at the interfaces
- Inadequate protection of data areas
- Inadequate protection against misuse of the permitted functions.

Thus in the specification what shall be specified is which mechanism are used for the separation. The evaluation team shall examine and rate these protection mechanisms carefully, many of which are realized by hardware or firmware. Using the specification, what shall be evaluated is whether these protection mechanisms are used adequately. Uncertainties or presumed vulnerabilities serve as the basis for the development of penetration tests.

In addition, the specification shall include why the security functions of the system components not to be evaluated may not be bypassed. What must be clear is that only the system components presented for evaluation possess the privileges required for the realization of the security functions. This shall also be substantiated by specific penetration tests performed during the evaluation.

Quality of the Software Development Process

Explanation of the Criteria

The evaluation of the implementation quality is restricted to the conduct of tests from the test library provided by the sponsor and such tests as were formulated for the testing of the security policy and the specification. These tests shall be adequate to indicate that the security functions listed in the specification are present and can be used in accordance with the specification and the documentation.

The following tests shall also be performed:

- The use of the security functions with illegal or meaningless parameter values
- The search for undocumented function (if the specification leads to the suspicion of their existence)
- The use of the security functions with parameters values located in the boundary area of the permitted parameter values.

If ambiguities concerning the specification are found (including functions not mentioned in the specification), either the specification and, if necessary, the security policy or the implementation have to be amended so that the consistency between security policy, specification and implementation is achieved. By means of such supplementary corrections, the evaluation effort is generally increased, as already evaluated components have to be reexamined. Thus such corrections should only be allowed to a slight degree.

Operational Behaviour

Explanation of the Criteria

All aspects, which are intended to guarantee enforcement of the security policy during operation, are covered by the term operational behaviour. These aspects vary widely, depending on the nature of the IT system. All relevant areas shall be defined and tested on the basis of the security policy for the system to be evaluated.

If different configurations have an impact on the security policy, this shall be reflected in the specification of the security functions. In addition, all configuration possibilities shall be documented in order to make the impact of different configurations clear to the user of the system.

It shall be possible to record interventions performed during the generation of the system. The non-deceivability of the audit is to be examined by appropriate tests.

In order to ensure that, when installing the software, no unrecognised errors occur, an procedure approved by the evaluation authority shall be used which can identify such errors.

Hardware maintenance and modifications to the software of the security functions are frequently areas in which the complete functionality of the security functions cannot be maintained continuously. In the evaluation of the system these areas are also to be examined and possibly be tested using specially constructed examples. As a result of these examinations organizational measures should be suggested to ensure a maximum of security even during maintenance.

The system has to have self-testing procedures for some hardware components in order to guarantee correct operation of the security functions.

Quality of the User-Oriented Documentation

Explanation of the Criteria

The quality of the documentation intended for the user is rated at the end of the evaluation. At this point in time, the evaluation team should have gained sufficient experience with the system to be able to assess the correctness, comprehensiveness and comprehensibility of the documentation. The sponsor must be notified of deviations between the real system behaviour and the documentation. Consequently the documentation must be improved before the certification takes place.

Assurance Level Q3

Explanation of the Criteria

Assurance level Q3 is intended for systems or system components of which moderate requirements are made as regards assurance of the enforcement of the security policy. The aim of the evaluation at assurance level Q3 is to demonstrate by the evaluation of the specification and selective testing of the implementation that the system is largely resistant to simple penetration attempts. Systems of assurance level Q3 are in many cases sufficient for areas with medium trust requirements regarding enforcement of the security policy. A moderate residual risk remains that sophisticated penetration attempts may reveal vulnerabilities in the system by means of which individual security functions might be bypassed or invalidated.

Quality of the Security Policy

Explanation of the Criteria

A verbal formulation of the security policy is still sufficient for assurance level Q3 too, but the use of semi-formal methods of presentation can facilitate the work of the evaluation authority and thus shorten it. In this context a careful evaluation means that (depending on the extent of the security policy) several members of the evaluation team must examine the document in detail and clarify any interpretation difficulties with the sponsor.

Quality of the Specification

Explanation of the Criteria

To achieve assurance level Q3 a detailed specification is necessary in which the implementation of the security functions and the other software is described which is not or is only inadequately separated from the security functions. For the evaluation of the system it is essentially this specification which is examined and the source code of the implementation is only involved in the case of uncertainties or presumed vulnerabilities. For larger systems the use of graphical representations is of great help, at least for the higher abstraction levels of the specification. Since the specification can still be informal, the trust placed in such an evaluation is naturally still relatively weak. It should however be possible to find most major design errors, even in such an evaluation.

Quality of the Mechanisms Used

Explanation of the Criteria

The rating "strong" should be the minimum rating which a mechanism shall have which has sole responsibility for the enforcement of a particular security requirement. This corresponds to the objective of assurance level Q3 which shall guarantee good protection against simple penetration attempts. A mechanism rated as "moderate" should only be used if at least one of the following pre-requisites is fulfilled:

- The mechanism is used in conjunction with other mechanisms. The combination of these mechanisms is rated "strong" or higher.
- The effort for the application of a stronger mechanism does not, in the opinion of the sponsor and the evaluation team, justify the costs incurred.

Quality of the Separation to the System Components not to be Evaluated

Explanation of the Criteria

The aspects mentioned in assurance level Q2 also have to be considered for the evaluation of the separation from the system components not to be evaluated when the tests are generated. In addition, for assurance level Q3 the implementation of the interfaces to system components not to be evaluated is subject to sample testing at source code level.

Quality of the Software Development Process

Explanation of the Criteria

For assurance level Q3 the presentation of the source code of the implementation of the security functions is required for the first time. For this reason requirements on the implementation language(s) are also made. However these requirements are still relatively weak. What is essentially required is that the languages used for the implementation are defined clearly. The evaluation team can require the sponsor to make the necessary documentation for the implementation languages available. This is particularly necessary when a not generally available language is used for the implementation. This also applies to preprocessors or other tools used in the code generation.

The penetration tests are selected basically according to the same criteria as for Q2. However, for an evaluation at assurance level Q3, uncertainties or presumed vulnerability have to be included, which were recorded during sample inspections of the source code. Due to the more careful evaluation of the specification, these tests are generally much more specific. An evaluation at assurance level Q3 is intended to certify broad resistance against simple penetration attempts.

For the evaluation at assurance level Q3 the sponsor must submit a library of test programs with the associated documentation appropriate to the number of the components of the system to be evaluated. These programs should be designed in such a way that the security functions are all tested with several parameter values. These tests shall also confirm the correct behaviour of the security functions. The library shall be designed in such a way that the evaluation team can easily rerun these test cases and create and run slightly modified test cases.

The requirements made on the implementation environment are still relatively weak at assurance level Q3. All that is required is that procedures shall be available with which the evaluation authority can generate all the components of the system to be evaluated from the source programs without any great effort. The version and alteration control may consist of creating only those parts when generating a new version of the system which were changed since the last version. Only for very large systems (this will be decided on in every single case by the evaluation team) must the manufacturer prove the separation of roles during the software development process as well as of a controlled integration and release procedure. This proof may be provided by presenting acceptance reports which clearly show who was responsible for the implementation (coding) and for the acceptance of the individual functional units. It should also be evident from these reports what tests from the test library were performed for this functional unit and when acceptance and integration took place.

Quality of the Operational Behaviour

Explanation of the Criteria.

The only new requirements for assurance level Q3 are for a secure initial state after a system start-up and the auditing of the system generation parameters. This is intended to prevent on the one hand that after start up the system can enter a state in which parts of the security functions are not enforced. At the very most this may only happen in certain maintenance cases and shall be audited. On the other hand the auditing of the system parameters should provide those responsible for the system with the possibility of reconstructing how their system was generated. This is particularly necessary when the evaluation only relates to certain generated variants and configurations of the system.

Quality of the User-Oriented Documentation

Explanation of the Criteria

The requirements made on the user-oriented documentation are identical to those required for assurance level Q2. The remarks there apply unaltered for level Q3 as well.

Assurance Level Q4

Explanation of the Criteria

Assurance level Q4 is for systems which have to satisfy moderate to high requirements as regards the assurance of the enforcement of the security policy. At this assurance level not only does an evaluation team analyse the specification, but the source code of the implementation is also subject to selective tests and sophisticated penetration attempts. This is intended to achieve a high degree of trust that the system enforces the security policy even in an environment in which well-meaning user behaviour cannot necessarily be assumed. Systems of assurance level Q4 are therefore suitable for areas with medium to enhanced requirements of trust in the enforcement of the security policy. There remains a relatively small residual risk that sophisticated penetration attempts might be able to exploit vulnerabilities in the system through which the security functions can be bypassed or invalidated.

Quality of the Security Policy

Explanation of the Criteria

To achieve assurance level Q4 the security policy on which the system to be evaluated is based shall be clearly more detailed than was described for assurance level Q3. Justification must be provided for each of the security requirements as to its purpose in the overall security policy, in other words what threat is to be averted or reduced by the fulfilled requirements. Graphical representations can be used to improve the understanding, e.g. of the mutual interdependencies of the individual security functions.

Quality of the Specification

Explanations of the Criteria

The use of semi-formal methods and the hierarchical structure of the specification are intended to provide the evaluation team with a better understanding of the algorithms and mechanisms used and their interrelationship.

The specification shall, in addition, be designed in such a way that the evaluation team can reconstruct the logic of the individual functions (such as identification and authentication of users), i.e. that no great freedom is given for the implementation of this function according to this specification. Thus ambiguities and vulnerabilities can already be looked for in detail in the specification. Therefore, this is particularly important, as the vulnerability analysis for an assurance level Q4 evaluation basically comprises a careful examination of the specification. The implementation is informally analysed in the evaluation at this assurance level, whereby ambiguities or presumed vulnerabilities resulting from the examination of the specification form the basis for the analyses. The realisation of the security requirements is reconstructed in the source code.

Quality of the Mechanisms Used

Explanation of the Criteria

It shall be explained which threat a mechanism used is intended to counter. All the threats listed in the security policy shall be covered by corresponding mechanisms. Each mechanism shall be described in such a way that an assessment is possible within the metric. This rating shall be undertaken for the evaluation of each of the mechanism used, whereby the categorisation of mechanisms listed in the IT catalog of criteria or in previous evaluations shall be treated as guidelines. Thereby the rating "strong" is the minimum rating for a mechanism.

Quality of the Separation from the System Components not to be Evaluated

Explanation of the Criteria

In order to achieve assurance level Q4, the separation from the system components not to be evaluated shall be performed very carefully. All interfaces to such components are evaluated carefully at source code level. In particular the correctness and comprehensiveness of the checking of parameters has to be taken into consideration. This includes an analysis of whether, after evaluation of their correctness, parameters can still possibly be modified by parallel, untrusted processes (time of check versus time of use (TOCTOU) problem). At level 4 higher demands are made of the quality of the separation mechanisms than at level 3. The minimum rating is "very strong" for separation mechanisms.

Quality of the Software Development Process

Explanation of the Criteria

If no officially evaluated compiler is used, the evaluation team will require the sponsor to provide a library of test programs with which the correct functionality of the compiler can be checked by the evaluation team. This library of test programs shall cover the possible language structures as far as possible. The nature and scope of the test programs can be specified by the evaluation team. It shall be possible for the evaluation team to run its own tests on all compilers used.

The use of an "exotic" programming language in the implementation may restrict the assurance level achievable. It cannot be assumed that the evaluation teams are totally familiar with every programming language. For this reason it is absolutely essential that a catalog of supported specification and implementation tools (which include programming languages and compilers) be issued. This catalog will of course have to be modified and supplemented as time goes by. In order to avoid wrong investments by a sponsor, what should be noted for each tool in this catalog is for which assurance levels it is appropriate and the minimum duration for which the evaluation authority will support this tool. If tools not listed in the catalog are used, the evaluation team can require from the sponsor that within the framework of the evaluation one or more persons of the evaluation team can be trained in the use of the tool.

As already mentioned for evaluations at assurance level Q4 an analysis of samples of the source code is required. The selection of these samples is left to the evaluation team and shall be oriented essentially to uncertainties or presumed vulnerabilities found during the evaluation of the specification. However it need not be restricted to these areas. In all samples the evaluation team shall evaluate how well the specification can be mapped on the source code. If components to be evaluated cannot be located clearly in the source code, or if the evaluation reveals major divergences between the specification and the source code, evaluation at assurance level Q4 is not possible.

Quality of the Operational Behaviour

Explanation of the Criteria

The evaluation of the impact of varying configurations can in most cases only be performed for selected test cases down to the source code level. Here too uncertainties or presumed vulnerabilities resulting from the evaluation of the specification shall form the basis for selection of the test cases.

The process of the system start-up should be analysed and tested carefully, so that it is certain that invalidation of security mechanisms by special interventions during the system start-up procedure is only possible for precisely specified exceptions. These exceptions (such as special maintenance) may only be available in a functionally restricted system in which manual supervision is possible. The same restrictions shall also be fulfilled if parts of the security functions are not active in other maintenance cases.

Quality of the User-Oriented Documentation

Explanation of the Criteria

Divergences between user documentation and real system behaviour frequently indicate vulnerabilities. Such divergences shall therefore also be subjected to tests down to source code level. After ambiguities are clarified, the evaluation team shall provide the sponsor with a chance to update the user-oriented documentation.

Assurance Level Q5

Explanation of the Criteria

Assurance level Q5 is intended for systems or system components of which high demands are made regarding the assurance of enforcement of the security policy. This assurance level is the first for which a formal security policy model is required. The aim of the evaluation is to demonstrate by careful analysis of the specification and implementation that the system is highly resistant to penetration. Systems of assurance level Q5 are suitable for systems with high requirements regarding the confidence in the enforcement of the security policy.

Quality of the Security Policy

Explanation of the Criteria

The formal model is intended to cover broad areas of the aspects of confidentiality and integrity (insofar as these aspects are significant in the security policy). The aspect of availability is generally very difficult or impossible to assess formally. Hence a formal model which covers all aspects of the security policy cannot be demanded here.

The proof of consistency for the model shall be submitted to the evaluation team by the sponsor at the start of the evaluation together with the remaining documents. The evaluation team cannot be expected to supply the proof of consistency during the evaluation. In addition the evaluation team shall be provided with all background information and tools used during the preparation of the proof. It is then the task of the evaluation team to follow and validate the proof steps. Moreover the verbally formulated security policy, which generally covers more than the formal model, shall be examined very carefully for consistency in itself and with the security policy model. Any parts not covered by the model shall be evaluated with particular care.

Quality of the Specification

Explanation of the Criteria

In order to be able to see the security model reflected in the specification with the appropriate level of quality, it is essential that a specification in semi-formal notation also exists in addition to the informal specification. These specifications shall be hierarchically structured and be sub-divided on each hierarchical level into clearly defined and basically independent functional units. A language with a clearly defined syntax has to be used for the semi-formal specification.

All the functions defined in the security model shall be exactly presents in the specification. It is virtually always essential for these to be defined in the specification as functional units as well.

The structuring at the uppermost hierarchical level shall be retained down to the lowest level, i.e. it may only be refined locally without the interfaces or linkages of functional units already in existence at higher specification levels being changed by this refinement. During the evaluation process the evaluation team will evaluate the specification at all hierarchical levels for consistency and enforcement of the security policy. If uncertainties occur or if vulnerabilities are presumed, these cases shall be recorded carefully since this is where the emphasis shall be placed for the evaluation of the implementation.

Quality of the Mechanisms Used

Explanation of the Criteria

For assurance level Q5 the requirements made on the mechanisms used are higher than for level Q4. Great care shall be taken in the rating of the mechanisms or the combination of mechanisms.

Quality of the Separation from the System Components not to be Evaluated

Explanation of the Criteria

From assurance level Q5, the examination of the separation from the system components not to be evaluated also contains the search for side effects which can be misused as covert channels i.e. by means of which information can be communicated in such a way that it violates the security policy. Covert channels do not only represent a risk in systems in which information is processed with hierarchical classifications but also generally in all systems in which read access to certain information is to be denied. Such covert channels can for instance also be misused in order to obtain authentication information, as the following example indicates:

In order to perform a certain function a system requires the input of a password. The password entered is compared byte by byte with the correct password. As soon as a discrepancy is established the function is interrupted and issues the error code x to the calling program. If the password to be entered is stored in such a way that only the first n characters are still in a memory area to which access is allowed, it is possible to decide using the error code whether the first n characters coincide with the correct password (as soon as the comparison function is about to access the (n+1) character, the program is interrupted with an error code y for unauthorized memory access). If the password is e.g. 8 characters long and each character from an alphabet with 36 possible characters, then by this covert channel the correct password is obtained on average after 144 instead of $1.4*10^{12}$ attempts.

This example also clearly indicates that the restriction of the bandwidth of a covert channel is not in itself sufficient in certain cases. In the case described restriction of the bandwidth of the covert channel to 1 bit/second would simply mean that on average it will take 64 seconds to determine the correct password. It is clear that such a channel cannot be tolerated. It is also necessary to check carefully what kind of information can be communicated via this channel, by which subjects the channel can be used, how great the effort for the exploitation of this channel is and how high the probability is that the exploitation of this channel is not noticed. The maximal bandwidth given in the criteria is intended for channels which cannot be eliminated without substantial restrictions regarding the availability of the system or the system functionality. The example described above, however, is a covert channel which can be eliminated without great effort and without restricting the availability of the system or the functionality of the system.

However the example also shows at what points of the interfaces to less trusted components of the system covert channels can appear. The error code can, under certain circumstances, reveal more information than is necessary and expedient in accordance with the security policy. Hence this is an example of an area which will be analysed particularly carefully by the evaluation team.

Quality of the Software Development Process

Explanation of the Criteria

During the evaluation at assurance level Q5, the evaluation team will also thoroughly analyse the source code of the implementation. Selected tests and informal control are no longer sufficient. Special attention shall be paid to the interaction between individual functional units and their interfaces during the evaluation.

Uncertainties or presumed vulnerabilities resulting from the evaluation of the security policy, the security policy model, the specification and the implementation are the starting points for selective penetration tests. These penetration tests can only refer to individual modules, whereby, for the selection of the input parameters, what can not be tested is whether this module will ever be called with these parameters in the system at all. The module shall behave correctly regarding its specification for all parameter values, since otherwise erroneous behaviour after a program modification cannot be excluded.

The use of a configuration management and control system is intended on the one hand to assure the consistency of the object code with the source code and on the other hand to allow to follow the development process of the product. Therefore the requirement of audit and separation of individual roles in the SW development environment is laid down.

Quality of the Operational Behaviour

Explanation of the Criteria

In order to be able to provide evidence of the enforcement of the security policy in the operational system in a form appropriate to assurance level Q5, including actual operation, the different configurations may only have a very light influence on the functionality. This generally means that for the configuration only a few constants can be redefined, but the principle logic remains essentially the same.

A new requirement for assurance level Q5 upwards is for trusted software distribution which is also intended to provide protection against tampering (manipulations). Cryptographic check sums or procedures of similar design are generally used here whereby the probability of non-identification of a manipulation if the manipulator does not know the key used can be determined precisely. A method approved by the evaluation authority for software distribution has to be adhered to.

The requirements regarding maintenance are tighter in that software maintenance may not involve any restriction of the security functions. This means that in the event of a modification to the actual security functions of the software, the system itself must enforce a state in which no threats external to the system are effective. This means, for instance, that in such a case no user may work on the system and all external links must be inactive. In addition special procedures shall be used which enable the manipulation of the software of security functions to be detected during a system start-up.

All requirements concerning operational assurance shall be evaluated by the evaluation team using examples, i.e. the evaluation team shall examine various configurations (if different configurations are possible), go through the maintenance of the security software as an example and cause or simulate system breakdowns in order to test secure restarting after such errors. The evaluation team shall function as the devil's advocate here and try to overcome the security functions of the system by malicious behaviour.

Quality of the User-Oriented Documentation

Explanation of the Criteria

The requirements made of the User-Oriented Documentation are the same as for assurance level Q4. In the case of ambiguities, the sponsor shall be given adequate time to make the necessary adjustments.

Assurance Level Q6

Explanation of the Criteria

Assurance level Q6 is intended for systems or system components of which high to very high requirements are made regarding the assurance of the enforcement of the security policy. In order to achieve this assurance level it shall be formally proven that the highest hierarchic level meets all the requirements of the formal security policy model. In addition the source code is analysed very precisely. The aim of the evaluation at assurance level Q6 is to obtain a very high degree of confidence that the system enforces its security policy and is very highly resistant to attempted penetrations. Systems of assurance level Q6 are suitable for systems with high to very high requirements as regarding the enforcement of security policy.

Quality of the Security Policy

Explanation of the Criteria

The formal security policy model shall be more comprehensive to attain assurance level Q6 than at the lower levels. All security requirements in the areas confidentiality and integrity shall be covered by the security policy model. Only the availability aspect, which is very difficult to formulate formally, may be omitted. However, the impact of the components not covered by the overall security policy on the overall security policy shall be analysed very carefully. The impact may not violate the axioms formulated in the formal model.

Quality of the Specification

Explanation of the Criteria

In order to achieve assurance level Q6 it is necessary to demonstrate formally that at least the top hierarchical level of the specification satisfies the formal security policy model. For this it is essential that this part of the specification be written in a formally defined specification language based on mathematical logic. In order to be able to prove the consistency between model and specification it is at least necessary for the language in which the model was written and the specification language to be carefully coordinated with one another. In most cases it will in fact be the same language.

The verification conditions to be drawn up to prove the consistency of the model and specification can hardly be drawn up and demonstrated for non-trivial systems without tools.

At any rate it is not possible for the evaluation team to reconstruct the completeness of the verification conditions without such supporting tools. Moreover the individual proofs are generally so extensive that they cannot be reconstructed in the context of an evaluation with the care due if no tools are available for this.

Although no formal verification of consistency with the highest level of the specification need be performed for the lower hierarchical levels of the specification, these levels shall also be written in the same specification language and the nature of the mapping from one hierarchical level to the next down shall be described formally. Thus theoretically verification down to the lowest hierarchical level of the specification would be possible, but this is not required for obtaining assurance level Q6. Here careful and non-formal evaluation of consistency down to the lowest hierarchical level is sufficient. The enforcement of the security requirements is conducted down to the source code.

Quality of the Mechanisms Used

Explanation of the Criteria

In conformance with the objective of assurance level Q6, "very strong" is the minimum rating which a mechanism may have for systems at this level. This generally excludes the necessity of major organizational measures for the effectiveness of the mechanism. It is also generally very difficult to upgrade a lower rated mechanism to "very strong" without any major design changes. Thus if it is established in the course of the evaluation that a mechanism does not have the necessary strength, what shall be clarified in discussions with the sponsor is whether the mechanism can be modified in such a way that it achieves the necessary rating, whether the evaluation should be terminated or whether a lower assurance level should be aimed for.

Quality of the Separation from the System Components not to be Evaluated

Explanation of the Criteria

In the case of an evaluation at assurance level Q6 the analysis of the interfaces to the components of the system not to be evaluated will be performed with special care down to the machine code level. Sophisticated penetration tests shall be used to try to misuse these interfaces to perform actions with which the security policy can be bypassed or invalidated. The machine code debugger shall also be used during these tests. Every error found in this way shall be repaired, since a known error which cannot be repaired leads to assurance level Q0.

The maximum bandwidth of covert channels is greatly limited at this assurance level.

Quality of the Software Development Process

Explanation of the Criteria

In order to achieve assurance level Q6 an careful analysis of the source code of the implementation is necessary. However such an analysis can only be performed by persons with experience in the application of the implementation languages used. This restricts the usable programming languages to those admitted by the evaluation authority, i.e. to those for which the evaluation authority possesses the actual know-how. At regular intervals, therefore, the authority will publish a list of the program languages admitted at assurance level Q6 and possibly of the compilers which can be used.

From assurance level Q6 onwards the evaluation also comprises a selective analysis of the machine code generated. This evaluation shall in particular cover those areas in which internal compiler mechanisms (such as e.g. the implementation of parameter passing to sub-programs, type checking treatment of range violation or treatment of errors) might under certain circumstances allow penetration attacks or the exploitation of covert channels. This implies that these areas (which also extend to the runtime system of the compiler) shall be documented very precisely.

The development and maintenance of the system must be monitored by a configuration management and version control system. Modification to objects, which are subject to the control of this system shall be audited.

Quality of the Operational Behaviour

Explanation of the Criteria

A very careful analysis of all configurations and their impact on the security policy is necessary to achieve assurance level Q6. This means an enormous effort when a large number of configurations are possible. For this reason the sponsor will generally only have the evaluation authority evaluate a certain number of these configuration possibilities (under certain circumstances perhaps only a single one). This is entirely feasible, but these configurations shall be described precisely in the evaluation report. The certificate then only applies for these evaluated configurations. The current configuration of the system shall be determinable at any time during operation by authorized roles.

A path approved by the evaluation authority shall be followed for the distribution of software. If security functions are invalidated during hardware maintenance then hardware maintenance may only take place with the explicit permission of the system administrator.

Quality of the User-Oriented Documentation

Explanation of the Criteria

The requirements made of the User Oriented Documentation are the same as for assurance level Q5. Given ambiguities between the real system behaviour and the user documentation, the sponsor is to be granted adequate time for improvements.

Assurance Level Q7

Explanation of the Criteria

Assurance level Q7 is for systems or system components of which extremely high requirements are made regarding the quality of the enforcement of the security policy. In order to achieve this assurance level it shall be formally proven that all the hierarchical levels of the specification and the source code of the implementation are consistent with the formal security model. The purpose of the evaluation at assurance level Q7 is to achieve an extremely high degree of confidence that the system enforces the security policy and is very highly resistant to penetration attempts. Systems of assurance level Q7 are suitable for systems with extremely high requirements regarding the assurance of the enforcement of the security policy. The effort for the preparation and evaluation of such a system is so high that with today's technical means this assurance level can only be realized for extremely small systems or system components with a very simple structure.

Quality of the Security Policy

Explanation of the Criteria

In order to achieve assurance level Q7 a formal model of the security policy is necessary which completely covers all individual aspects of the security policy. This means that no security requirements may be demanded which cannot be represented by a formal model. This itself restricts the systems which can be evaluated at assurance level Q7 with today's technical means.

Quality of the Specification

Explanation of the Criteria

In order to achieve assurance level Q7 evidence shall be furnished of the consistency between the formal security policy model and the specification down to the lowest hierarchical level of the specification. Only in this way is it meaningful to demonstrate the consistency between the lowest specification level and the program source, i.e. code verification. That the security policy is enforced is proven down to the source code. However as for assurance level Q6, this presupposes that only methods and tools approved by the evaluation authority have been used. Otherwise it is not possible for the evaluation team to reconstruct the completeness and correctness of the evidence furnished by the sponsor.

Quality of the Mechanisms Used

Explanation of the Criteria

In accordance with the assurance requirement of level Q7 only mechanisms rated "virtually unbreakable" are used. Mechanisms rated "very strong" shall only be used where it is not possible to realize a mechanism rated "virtually unbreakable" with the technical means available today.

Quality of the Separation from the System Components not to be Evaluated

Explanation of the Criteria

The only differences to assurance level Q6 are the higher requirements concerning the assurance of the separation mechanism and the further restriction of the maximum bandwidth of a covert channel allowed. No other aspects over and above assurance level Q6 apply for the evaluation of this component.

Quality of the Software Development Process

Explanation of the Criteria

The greatest differences between the assurance levels Q6 and Q7 relate to the evaluation of the source code of the implementation. In order to be able to perform a verification down to source code level, the programming language used shall possess formally defined semantics in all respects. Generally, however, it is not possible to define all points of the semantics independently of the target hardware. That is the reason for the requirement that such points shall be formally defined in the documentation of the compiler used. The machine code generated shall also be carefully analysed at assurance level Q7. In order to be able to reconstruct the mapping between the source code and the machine code manually with the necessary degree of precision, the compiler may not perform any complex optimizations. The compiler shall also be able to generate a listing which represents source code and generated machine code in a form which permits an easy and clear mapping between these two. The machine language of the target hardware used shall also be formally defined to a large extent. All test cases from the test library will be run and shall generate the results documented.

The search for covert channels shall be performed with extreme care. If a covert channel found cannot be eliminated, a careful analysis shall be made of the manner in which it can be exploited and the type of information which could be revealed via this channel. It shall be then carefully considered whether the presence of such a covert channel still permits classification at assurance level Q7. Only cleared personnel may participate in the development and maintenance of such systems.

Quality of the Operational Behaviour

Explanation of the Criteria

The only essential difference from the criteria for assurance level Q6 are the selected tests of the recovery procedures. For this the evaluation team shall attempt selectively to cause or simulate particular errors which could lead to a failure of the system. The recovery programs become part of the system components to be evaluated if they temporarily bypass or invalidate certain parts of the security policy. This means that it shall be verified that these programs are correct regarding their specification and that upon termination of the recovery the system is in a secure state.

Quality of the User-Oriented Documentation

Explanation of the Criteria

No new criteria have been added for assurance level Q7.

4. Explanations of the Documents Required

The following explanations are intended to provide an overview of what shall be included in the documents which describe the requirements at the individual assurance levels. The actual content (scope and degree of detail) of these documents depends to a large extent on the system to be evaluated and the assurance level aimed for.

It is not necessary that all the descriptions applicable to a topic or to a particular sub-topic shall be included in one single document. References to other documents are permitted, whereby attention shall be paid to the fact that the readability of the documents may not suffer from too many references.

In generally valid is that the evaluation team decides in all cases of doubt on the of the type of presentation, contents and comprehensibility of documents.

Here explanations of the requirements at assurance levels Q1 to Q3 are given. At the higher assurance levels, these explanations are equally applicable: however, in part, more stringent requirements also shall be taken into account (detailed presentation, presentation in semi-formal or formal relation etc). More precise explanations of the requirements from assurance level Q4 onwards will be provided in a later version of the IT Evaluation Manual.

The following documents are required in the IT security catalog for an evaluation at the assurance levels Q1 to Q3:

- Description of the security policy.
- Specification of the system components to be evaluated.
- Description of the separation from the system components not to be evaluated and the interfaces to these components.
- Documentation for the user, i.e.
  - description of the application of the security functions, divided into the roles defined according to the security requirements.
  - description of the security-relevant aspects for system generation, system start-up, system administration and system maintenance.
- Description of the hardware and firmware used with presentation of the functionality of the protection mechanisms realised in the hardware or firmware.
- Test documentation (from Q2).

4.1 Description of the Security Requirements

What are Security Requirements?

The security requirements define which security functions are demanded of an IT system. They form the basis for the evaluation, in which is demonstrated whether the security functions of the IT system fulfil the security requirements.

Normally a system has additional functions. If these functions are not identified in the security policy as being security relevant, they are not tested for such so long as they do not have any impact on the security functions to be evaluated.

The security requirements is usually formulated by the manufacturer of a system. As however, it is possible that a user commissions the evaluation it is feasible that the user defines the security requirements and thus determines which functionality of the system shall be tested in the sense of its security requirements.

Explanation of the Requirements in the IT Catalog of Security Criteria

What shall be described in the security requirements is which security function and sub-functions a system or individual component contain.

A reference to a particular class of functionality is insufficient, the description must be more detailed.

Example: In F2, what is required is that the system administers access rights between subjects and objects. Thereby no statement is made which objects are subject to the administration of rights (e.g. volumes, files, library elements, record etc). The precise definition of the protected objects has therefore to take place in the security requirements.

Therefore a detailed description of the security requirements is required because a system must not necessarily exactly fulfil the requirements of one or more classes of functionality. It is also feasible that a system only fulfils particular security requirements which have not been included in any of the previously defined functionality classes in that combination or that the system fulfils the requirements of one or more functionality classes and several further requirements in addition.

From Q2, it shall be clear from the security requirements which threat or threats is/are to be averted by the individual security functions and to which basic security function(s) (identification and authentication, administration of rights, etc.) the individual implemented security functions apply.

It is left to the manufacturer to select the form of presentation (listing of points, informal description, graphic). It must however be clear which security functions are fulfilled.

4.2 Specification of the System Components to be Evaluated

What is a Specification?

A specification shall describe the realisation of the security functions of an IT system completely and comprehensibly.

For complex systems the system shall be made up of several descriptive levels (hierarchic levels), as the security functions can only be described comprehensibly in this manner. In cases of doubt the evaluation team decides on the necessity of a multi level description. On the top descriptive level, generally the "what" and "where" can be found (i.e. functional description, breakdown of the system into individual functions, etc.); on further levels, this rough outline is refined (structured) according to the definitions in the Duden-Fremdwörterbuch [2], i.e. here the "how" is described (e.g. algorithms, control flow, data, details of the implementation etc).

The terms for the individual descriptive levels of the specification, have not been standardized. Not even the term "specification" is applied uniformly.

[2] Duden-Fremdwörterbuch

Spezification
1 Categorization of the types
2 Individual listing

to specify
1. to list individually, to enter (on a list)
2. to structure

The top descriptive level is sometimes named as in the IEEE definitions [3], "Design", the lowest level (and the middle ones, if such exist) are known as "design specification".*)

In the IT security criteria catalog "specification" is the description of the system over all the hierarchical levels.

It is not necessary that different documents exist for the various descriptive levels of the specification, so long as it is possible to filter out the information relevant to the individual descriptive levels. The evaluation team decides on the presentation permissible. Under the same conditions, the strict separation between "what" and "where" (design) on the one hand and "how" (design specification) on the other hand is not imperative. A structure or the documentation which conforms to the above definitions will improve the comprehensibility in general and therefore facilitate the verifiability.

Explanation of the Requirements in the IT Security Criteria Catalog

Contents and scope of the specification depend on the assurance level aimed for and on the complexity of the system to be evaluated. Thus the formulation in Q2 and Q3 "if the specification is hierarchically structured" is to be understood as follows:

An IT system is usually too complex to allow for the possibility to describe it precisely with only one hierarchic level that the mapping between security requirements and source code (in Q3) can be so followed. In such a case, the description is absolutely essential at several hierarchic levels.

[3] IEEE STD 729-1983

Design:
The process of defining the software architecture, components, modules, interfaces, test approach, and data for a software system to satisfy specified requirements.
The result of a design process.

Design specification:
A specification that documents the design of asytsem or system component; for example a software configuration item. Typical contents include system or component algorithms, control logic, data structures, data set-use information, input/output formats and interface descriptions

Specification:
A document that prescribes, in a complete, precise, verifiable manner, the requirements, design, behaviour, or other characteristics of a system or system component.
The process of developing a specification.
A concise statement of a set or requirements to be satisfied by a product, a material or process indicating, whenever appropriate, the procedure by means of whichg it may be determined whether the requirements given are satisfied.

The description shall have sufficient hierarchic levels so that an understanding of the structure of the security functions can be obtained and, depending on the assurance level aimed for, also for the internal structure.

At the highest hierarchical level of the specification a general description of the overall functionality is required. This description shall show how the security requirements are mapped onto functions. For a small system or for a small individual component, the algorithms applied can be described here, for a more complex system, this has to happen at a lower hierarchic level.

Furthermore, what has to be described is how the distribution or the security functions is undertaken to the individual functional units. "Functional units" can be depending on the assurance level aimed for and depending or the current hierarchic level, complex system components which fulfil a security function (Q1, Q2) or groups of modules, modules, procedures, routines, elementary functions etc which fulfil subfunctions (Q2, Q3).

The user interface and thus the interfaces of the security functions visible at the user interface shall be described in full. From Q2, onwards "users" are also internal functional units; therefore also the internal interfaces of the functional units shall be described.

The control and data flow between the system and its environment and between the individual functional units shall be described precisely, beginning with Q3 down to the level of modules, procedures, routines and elementary functions.

Beginning with Q3, the control and data flow with in the individual functional units and their internal data structures shall be described in such detail that a mapping is possible to the source code.

Beginning with Q2, particular emphasis shall be placed on the description of parameter validation, privileges examination and error treatment.

A listing of minimum requirements on the specification now follows. **It is emphasised again that at higher assurance levels the specification shall be more detailed and more exact, even if no additional requirements have been made.**

Besides these requirements of the specification, further information may be necessary to understand the system (e.g. specific hardware descriptions, particularities of the implementation). The evaluation team determines in cases of doubt which further elements of the specification shall be submitted.

Requirements on **Functionality and Structure**

☞ From Q1:

In principle: The internal structure shall only be outlined roughly.

- Description of the task and impact of every functional unit and its contribution to the security functions.
- Description of interaction and dependences of several functional units.
- Description of the operation and the algorithms.
- Description of the implementation of the security functions.
- Description of the constraints which are necessary for the understanding.
- Description of the testing of special rights and privileges.
- Description of particular aspects of individual functional units, which shall be explained in the user documentation.

☞ Additionally from Q2:

In principle: The internal structure shall be presented in detail.

- Description of the call hierarchy.
- Description of particular characteristics of the functional units (resident, reentrant, serial reusable, etc.).
- Description of the serialisation and synchronisation mechanisms used.

Requirements on **Data and Parameters**

☞ From Q1:

- Description of the memory areas used and their attributes (e.g. memory protection attribute).

☞ Additionally from Q2:

- Description of the parameter checking (validation) to call interfaces.
- Description of global data structures which are important for the understanding.

☞ Additionally from Q3:

- Description of local (i.e. specific to a functional unit) and or global data and their structure.
- Description of the accesses and access types to data structures.
- Description of the access paths used to reach global data structures.

Requirements on **Interfaces**

☞ From Q1:

- Description of the call interfaces and the data transmitted.
- Description of the necessary privileges of the caller.

☞ Additionally from Q2:

- Description of the internal interfaces between individual functional units.
- Description of the functional units called.

Requirements on the **Error Recovery**

It is assumed that the system contains mechanisms which can counter a variety of internal errors e.g. (invalid operation code, addressing of non-existent memory areas) without impairing the overall functionality. If further requirements concerning error recovery and guaranteeing the functionality exist, these shall be defined in the security requirements.

☞ From Q1:

- Description of the error recovery for erroneous input parameters.
- Description of errors and events which shall not occur and how this is prevented.

☞ Additional from Q2:

- Description of error recovery for internal errors.

NOTE:
The previously mentioned requirements apply not only to functional units which contribute to security functions but also - in as for as applicable - to all IT system functional units, which are not sufficiently separated from the functional units which realise security functions (see also Chapter 4.3).

4.3 Description of the Separation of the System Components not to be Evaluated and the Interfaces to These Components

Why does the separation have to be described?

Assuming a security function is virtually unbreakable, if there is a way of compromising the system by bypassing this security function, then the security function is basically worthless.

Example: On access to a file using its name, the validity of the access by this user is tested. However, it is also possible for certain users to bypass the evaluated component and to read directly the individual sectors of the disk on which the file is stored and thus, of course, the information in the file.

This examples illustrated why during an evaluation the separation mechanisms, to the system components not to be evaluated shall be tested in addition to the separation mechanisms to the system environment.

For further details please see Chapter 5 (Explanations of Separation).

Explanation of the Requirements in the IT Security Criteria Catalog

The requirements listed in Chapter 4.2 concerning the specification of the system components to be evaluated apply analogously also for the mechanisms which realise the separation and the interfaces between system components to be evaluated and those not to be evaluated.

In particular attention shall be paid here to the fact that the description of the separation mechanisms is complete and that an explanation is given why these cannot be bypassed.

4.4 Documentation for the User

Explanation of the Requirements in the IT Security Criteria Catalog

The requirements listed here refer to the functions concerning the security of the system, whereby user is meant to be the normal end user and the system administrator.

The documentation shall convey such detail of the security functions of the IT system which concern him to every user that he is capable of applying these functions without error.

All user interfaces of the security functions shall be described with their parameters. If dependencies to other functions exist, these shall be described as well. In particular, the user shall be made aware of the possible consequences, of e.g. particular parameter combinations which are not perhaps visible at first sight.

The security problems during generation, installation, start-up and maintenance of the system shall also be dealt with in the relevant documentation.

4.5 Description of the Hardware and Firmware Used with the Presentation of the Functionality of the Protection Mechanisms Realised in the Hardware or Firmware

No explicit requirements are made on this point in the IT security criteria.

The general IT security criteria philosophy applies that, given ascending assurance levels the requirements on quality and detail of the presentation increase.

Explanation of the Requirements in the It Security Criteria

The description of hardware and firmware is necessary for understanding the specification of components interfacing with the hardware and also for the rating of mechanisms and the assessment of the separation to system components not to be evaluated (e.g. storage protection mechanisms, ring architecture). It must therefore contain the information necessary for this.

In many cases, the relevant documentation of the hardware manufacturer is applicable.
The documentation must at least contain:

☞ From Q1:

- Instruction set of the processor.
- Description of the most important architectural characteristics (e.g. system states, memory protection).

☞ Additional from Q2:

- Description of important peripheral control units.

☞ Additional from Q3:

- Complete description of the hardware architecture.
- Description of all peripheral control units.

4.6 Test Documentation

Requirements concerning the test documentation can be found, in the section "quality of the software development process".

Explanation of the Requirements in the IT Security Criteria Catalog

All information shall be included in the system tests documentation which is necessary for the reproduction of the individual tests by the evaluation team. In addition, this documentation shall include the results generated during the conduct of the tests.
All test programs or test procedures and their input data shall also be available on magnetic media.

The information and data to be documented are at least:

- Hardware configuration (with version and revision numbers).
- Software configuration (with version and revision numbers).
- Test plan and test goal.
- Test method.
- Test program or test procedure (with version and revision numbers).
- Input data for the test programs or procedures.
- Special features and dependences.
- Test result.

5. Explanations on Separation

This chapter contains explanations to the quality aspect listed under "Quality of the Separation to the System Components not to be Evaluated" in the quality assurance levels.

In an evaluation at first it will be assumed that the IT system to be evaluated can be sub-divided into system components to be evaluated and those not to be evaluated.

All the components of the IT system are meant by the term "system components to be evaluated"

1. which realise security functions.
2. which perform necessary system services for security functions.
3. which are not adequately separated from 1 and 2.
4. which realise separation mechanisms.

The system components not to be evaluated comprise all those IT system components

which do not perform security functions,

which do not support security functions,

which do not realise separation mechanisms,

which are sufficiently separated by respective mechanisms (this depends on the quality assurance level aimed for) from the system components to be evaluated.

At the beginning of an evaluation, it is in general unclear how the exact partioning of the IT system in system components to be evaluated and those not to be evaluated looks like. Therefore, the manufacturer of the IT system has to submit a document in which is described, from the manufacturers viewpoint, how the system components to be evaluated and those not to be evaluated are separated, which separation mechanisms are used and which interfaces exist to the system components not to be evaluated. A first partioning of the IT system is gained from this document. The final partioning is only achieved in the course of the evaluation.

An important criterion for the assessment of the quality of an IT system is the strength of the separation mechanisms. In the course of the evaluation, what is examined are

- whether the security functions, which are to be realised by the system components to be evaluated, can be bypassed by other components in the system.

- whether the security functions can be deceived by other system components.

- whether the security functions can be misused by other system components so that this represents a violation of the security policy.

What emerges in many cases is that there are components in the system which do not actually contribute directly to the enforcement of the security policy, however are not separated or are inadequately separated from the security functions. As it cannot be excluded that these components influence the functionality of the security functions, these components will also be evaluated.

Examples of Separation Mechanisms

1. Separation Via Different Privilege States

   Many processor families provide different privilege states. In most cases, these privilege states are hierarchically ordered, where a successive smaller subset of the total instruction set can be executed in the lower privilege states.

   For simplicity, we assume a processor with only two privilege states, called the privileged and non-privileged state. In the non-privileged state all I/0-instructions, instructions which modify special status or control registers and instructions changing the memory protection attributes are disallowed. A specific instruction switches the processor from the non-privileged to the privileged state. An operation system can use this mechanism to prevent an application program from directly accessing peripherals like disks, printers or communication lines. To do this in a secure way, the operating system must protect its code and data structures from unauthorized modifications by application program. This implies a memory protection mechanism combined with the privilege states (access to specific areas of memory is only allowed, if the processor is in the privileged state).

   This mechanism may be used, to provide specific operating system services for programs executing in the non-privileged state. Such a program may pass parameters to the operating system service (by putting them into specific areas of memory or specific registers) and then execute the instruction, which changes the processor from the non-privileged to the privileged state.

   To separate the operating system from the application software the following rules shall be obeyed:

   - Every instruction executed in the privileged state shall be protected against modifications from application programs.

   - Every data structure used by the program running in the privileged state shall either be protected against unauthorized modification or checked for correctness and consistency before use. For those data structures unauthorized changes between the time of check and the time of use shall be prohibited.

A violation of one of those two rules makes it in many cases possible to penetrate the system. If no additional firmware support is given, software shall enforce both rules. In some systems, this results in a large and complex program, which has to deal with several possible side effects (e.g. some parts of the program must not be interrupted). Some modern processor families provide a very good firmware support, which help to enforce the two rules. This may be an automatic context switch, making a copy of the parameter list or even switching to another address space.

In this example the consequences for the evaluation are: Every program which may execute in the privileged state is subject to evaluation.

2. Separation by Virtual Address Spaces

Virtual address spaces are also a mechanism to separate security relevant from non security relevant software components. For example some security services may be provided by a program running in a special virtual address space. The strength of this separation depends on several factors:

- The protection of the page tables. If those tables can be manipulated by an untrusted program, the separation mechanism is very weak.

- The existence of common code or data areas. If code or data areas of the security relevant software are mapped into a virtual address space containing untrusted software, it shall be checked, if those code and data areas con be read or modified in an unauthorized way. This kind of common areas will normally increase the evaluation effort dramatically.

- The protection of the communication features between different address spaces. Complex communication features will normally also increase the evaluation effort.

6. Evaluation Environment

Besides the technical rules for the evaluation procedure, legal and organisational constraints will be defined for the evaluation and rating of IT systems or IT components.

The Federal German Cabinet approved the draft of a law for the setting up of a Federal Office for Security in Information Technology - GISA will become a Federal Office - on 21st February 1990. The law and the supplementary decrees envisaged will include the necessary legal regulations, in as far as the Federal Office or Offices commissioned to act on its behalf become active. If required, supplementary organisational regulations will be included in the evaluation manual.

7. Description of the Evaluation Process

General Remarks

The following chapters provide a wide variety of remarks on how the evaluation of a system should proceed, what organizational measures should be taken and what mode of operation appears to be adequate. These are remarks, as already stated, and are not meant to be seen as rigid rules.
These remarks indicate that considerable organizational effort is required for an evaluation. This is justified all the more if the system to be evaluated either aims at a high assurance level or is sufficiently complex. However, this warning should be noted: Evaluators should not go into detail right from the start and hope to spare this organizational preamble.
In the course of a number of evaluations sufficient experience will be gained to be able to review the remarks made here again.

7.1 Organizational Structure of the Evaluation Team

An evaluation team comprises:

- The project manager (responsible for the project).
- The technical project leader.
- The evaluators.
- The moderator.

The tasks and responsibilities of the individual members of an evaluation team are described in more detail here.

PROJECT MANAGER

The organising project manager is responsible for the overall conduct of the evaluation. He must ensure, as the person responsible for the project, that the schedule of the evaluation is adhered to and the costs of the evaluation do not exceed the specified framework. His tasks include:

- The structure of the contract.
- Participation in the evaluation planning.
- Composition of the evaluation team.
- Participation in reviews during the evaluation.
- Reporting.

The project manager defines, in conjunction with the manufacturer, the product to be evaluated, the functionality aimed for or the class of functionality and assurance level (evaluation target) and documents this in the evaluation contract. As the person responsible for the project, he participates in the evaluation planning. The organising project manager reports to his superior on the status and progress of the evaluation and on problems which endanger the evaluation process from the viewpoint of the evaluation team.

He also informs the sponsor and the evaluation authority of current problems which endanger the progress of the evaluation. In addition he notifies the sponsor about the time scales for improvements and of inadequate resources.

TECHNICAL PROJECT LEADER

The technical project leader is responsible for the technical conduct of the evaluation. He distributes the current tasks to the evaluators. He thereby takes the specific knowledge of the individual evaluators into account. His tasks include:

- The technical project planning.
- Active participation in the evaluation.
- Reporting.
- Decisions on technical problems.

The technical project leader decides on the technical aspects of the evaluation. He makes amendments to and refinements in the project plan which become necessary in the course of the evaluation. The tasks defined in the project plan are distributed by him to the individual evaluators. The technical project leader is to be viewed as a normal member of the team in all technical review meetings with all the rights, duties and responsibilities. More over he should take part in all other meetings which concern the project, in order, on the one hand, to present all the technical items of information to the project manager and, on the other hand, to inform the team members of decisions made by the project manager and to be able to explain them. In addition, he reports all problems to the project manager, as they occur during the evaluation. He makes decisions in all the problem cases which do not affect the time schedule of the evaluation or the evaluation target.

EVALUATORS

The performance of the evaluation is their responsibility. Their tasks include:

- The assessment of the individual documents such as a specification, etc.
- The development of test programs.

- The development of test data sets.
- The conduct of tests.
- The assessment of the user documentation.
- The compilation of documents for all evaluation steps.

In addition, the evaluators report to the technical project leader on problems which have occurred during the evaluation and the premature termination of a sub-evaluation. They present and substantiate their test results in sub-evaluations as part of a review meeting.

MODERATOR

The moderator is responsible for the planning and conduct of reviews. His tasks include:

- The planning of review meetings.
- The chairing of review meetings.
- The reporting.

The moderator determines the time and place of review meetings. He names the persons who will participate in the review (in general, these are the evaluation team members), he prepares the actual review, distributes the working documents and notifies the review participants in good time as to time and place of the review. His tasks includes the chairing of the review meeting so the moderator should be experienced in this field. He takes the minutes of the review meeting and prevents unproductive discussions between participants in the review. In addition he supervises the time schedule of a review meeting and defines work packages which result from the review. Review results and the problems which occur are passed by the moderator to the technical project leader and on to the project manager.

The tasks of the moderator can, if required, be taken on alternately by other members of the evaluation team.

7.2 The Review Process

At the end of an evaluation a decision must be taken as to whether the evaluated system is to receive a certificate or not. This decision must naturally always be seen in relation to the criteria to be satisfied. Since a multitude of criteria are to be satisfied, the final decision is made up of a large number of individual decisions.

It must also be noted that the requirements specified in the catalog of criteria are not physically measurable quantities. The goal of the formulation was of course to allow objective statements when the criteria are applied. Despite this a certain residual subjectivity always remains. The experience of the evaluator is of crucial importance here.

In order to keep the influence of these subjective decisions on the overall process as low as possible, the following procedure is recommended for the evaluation process.

The evaluation process is subdivided into individual phases and these in turn into individual steps. Within these individual steps the evaluator must process individual work packages. If this work package specifically prepares a decision, as to whether a criterion is satisfied or not, this decision is not left to the individual assigned to this work package. His task is to prepare a decision. If a work package is completed from the point of view of the evaluator, i.e. if sufficient knowledge is available, he requests the moderator to convene a review session.

In this session the evaluator explains his mode of operation, the tasks/tests performed, problem areas and his suggested result of the evaluation. In order that those attending (technical project leader, evaluators and possibly the project manager) can come to a decision, the moderator must have presented them with a rough outline of the work done and the conclusions reached to be discussed at the review session. The evaluator is responsible for the preparation of this document.

Under the chairmanship of the moderator those present discuss the findings presented with the aim of reaching a joint decision as to whether the suggested evaluation result can be accepted in this form or whether uncertainties remain which necessitate additional investigations.

The aim should be a unanimous decision if possible. However the possibility of one participant not being able to support the decision cannot be excluded. In this case his arguments shall be recorded and added to the final report for this work package.

An evaluator can also request the moderator to convene a review session if he comes to the conclusion that a given criterion is not satisfied and further investigations are no longer meaningful. Under these circumstances it is also the aim of the review session to reach a common understanding. If the majority of the participants reach the conclusion that there is still a need for further investigations, the arguments leading to this must be recorded.

It is then left to the discretion of the technical project leader and the project manager whether they follow these arguments and advocate further investigations. If the project leader and the project manager differ in their opinions, the final decision rests with the project management who also bears the overall responsibility.

7.3 How to Start an Evaluation

In principle there are two ways to start an evaluation.

Firstly by a manufacturer who is offering a product, or

secondly by a potential user of a product.

The interest of the manufacturer consists in having the assurance of its product fulfilling certain security requirements established by an independent organization. The interest of the user consists in discovering whether a product which appeals on purely functional grounds also enforces his security requirements. In addition he must determine the desired assurance level, since this has an influence on the residual risk if he operates this product in his environment.

The preliminary work which both must perform before they submit an application for the evaluation of a product is described in further detail below.

Preliminary work by the manufacturer:

In the case of an existing product the manufacturer can generate a large number of potentially enforceable security requirements from the given security functionality of its product. On the one hand he will try to formulate the maximum number of security requirements which can be enforced by his system. On the other hand it may be possible that various configurations of the system can enforce different security requirements, and possibly with different degrees of assurance. If the manufacturer does not already have a potential user in mind whose security requirements he knows, it is also conceivable that he could follow the security policy described in a class of functionality of the national catalog of criteria. The catalog of criteria contains a chapter on classes of functionality in which the first classes describe the security requirements as required in the "Orange Book" classes (C-A). The description is built up in such a manner that it is oriented towards the defined basic security functions and allocates the individual security requirement to the appropriate basic security function. The manufacturer must decide clearly on a particular set of security requirements for the enforcement of which the system is to be evaluated.

The individual assurance levels indicate to the manufacturer which criteria his software must satisfy during development, maintenance etc. This helps him to assess which level the system to be evaluated can reach at most.

Thus in a request for an evaluation he can already inform the evaluation authority of a class of functionality and assurance level at which he would like to have his product evaluated. It is the responsibility of the manufacturer to approach the evaluation authority with realistic requirements. If the manufacturer cannot give exact details concerning the security function of his product, the following is suggested:

The manufacturer has his product pre-evaluated, at his expense, by independent external consultants with evaluation experience and, possibly, an impartial observer from the evaluation authority (no official participant). The aim of this pre-evaluation is to achieve a binding statement on the functionality of the product's security functions. Such a pre-evaluation should not last longer than two weeks. The manufacturer should go to the evaluation authoring or to an authorized evaluation office when selecting his possible external consultants.

A pre-evaluation is however, in general only possible for the functionality or class of functionality of a product. Realistic evidence on the assurance level achieved cannot, in most instances, be made in such a short time. At best, only indications can be given.

Preliminary work by the user of a product:

The user looks for a likely product to fulfil his operational requirements. By assessing the threat arising from the environment the operational functionality and other influencing factors, he elaborates his security policy together with the assurance level appropriate to the threat.

The resulting assurance level naturally has an impact on the form of presentation of his security requirements. For assurance level Q5, for instance, a formal security requirements model is to be prepared and presented to the evaluation team. The resulting security requirements need not correspond to a given class of functionality. However they form the basis on which the evaluation team evaluates the system. Since the assurance level to be achieved naturally depends not only on the form of presentation of the security policy but also on the assurance achievable by the rest of the system, the user would be well advised to contact the manufacturer of the system at an early stage. In these talks the manufacturer shall indicate to the user the potential assurance level which his product can achieve in an evaluation. At the same time what should be clearly noted at this stage is whether the security policy outlined by the user is at all enforceable from the point of view of the manufacturer. If agreement is reached, the manufacturer and the user will formulate the security policy in the form required by the assurance level found to be desirable and achievable. The user today is hardly in a position to prepare a formal model of his security policy by himself. It also does not make sense to present the security policy in a form which goes far beyond the requirements which the software product requires on the basis of the assurance level achievable by its design. The manufacturer shall of course also be prepared to make available the relevant documents necessary for the assurance level to the evaluation authority and to draw these up if they do not yet exist. Once all these points are clarified, the product can be submitted for evaluation for this assurance level together with the security policy. From here on there is no difference between the evaluation initiated by a manufacturer or by a user.

7.4 Sequence of an Evaluation

Phase 1 ESTABLISHING CONTACT

Stage 1

The sponsor establishes contact with the evaluation authority. He presents his wishes in an application for evaluation.

Stage 2

Discussion of the request for evaluation i.e. above all, is the time schedule acceptable to the evaluation office. Further information can be obtained in discussions with the applicant in order that the following list of points can be dealt with and be answered.

- Determining the points of contact.
- Determining the personnel requirement.
- Determining how many external staff are required.
- Details concerning the earliest starting date.
- Details as to whether the number of personnel made available by the manufacturer is sufficient.

Stage 3

Specification of criteria, i.e. functionality and assurance level according to which the product is to be evaluated. The list of documents and objects to be handed over derives from this. The assurance level defines how the objects are to be handed over (source code, load module, etc.). If special hardware is required, agreement of when this shall be delivered. Clarification of liability for hardware and maintenance questions. Specification of time frame to be aimed at for the individual phases for the review with the associated resources. Determination of personnel required with all concerned. Specification of training measures.

Stage 4

Elaboration of the contract with sponsor and external support firms. Components of the contract are all the documents elaborated in stage 3, such as list of documents etc. On conclusion of the contract the object to be evaluated is specified, as is the associated documentation. An exception is made by the accompanying evaluations (see Chapter 8).

During the evaluation the sponsor is not permitted to bring any altered documentation or an altered component into the evaluation process. This is only possible with the agreement of the evaluation authority represented by the project manager.

Phase 2 EXAMINATION OF DOCUMENTS

Stage 1

- Acceptance of all objects and documents.
- Elaboration of the first work plan.
- Assignment of staff to rate the assurance criteria (Q group).
- Assignment of staff to rate the functionality criteria (F group).
- Allocation of documents to the 2 groups and work packages.
- Agreement on the first review (time scheduling).
- Agreement on tools to be used jointly (test preparation).

Stage 2

Q group checks whether for the assurance level aimed at

- all necessary documents are on hand.
- form of presentation is sufficient on a superficial examination also, with reference to identifiable components.

Setting up of a reference list: Which document contains statements on

- security policy.
- specification of the security functions.
- mechanisms.
- software development process.
- separation from system components not to be evaluated.
- operational behaviour.

(It cannot be expected that the documentation supplied is structured in accordance with the catalog of criteria.)

The F group checks the security requirements and abstracts necessary basic security functions as well as individual requirements.

Does documentation for basic security functions and individual requirements exist?

Setting up of a reference list: which documents contain statements on the identified basic security functions and individual requirements.

Stage 3

From the individual reviews of stage 2 an overall document is to be drawn up which addresses the following points:

- Can the evaluation be continued with the specified criteria?
- Where can critical areas occur?
- Correction list.

Stage 4

Handing over of this document to the sponsor with agreement on

- What can be corrected by what date?
- Impact on time schedule and assignment of personnel.

Stage 5

Work packages of stage 1 to stage 3 are performed once again for the corrected/improved documents.

Stage 6

Final decision on whether the evaluation is to be continued and whether previous criteria are still valid.
If the evaluation is terminated:

- Prepare final document justifying termination.
- Dissolve contracts with supporting firms.
- Return objects and documents.
- File documents prepared in the process of the project.

If the evaluation is continued:

- Specify the documents, objects and evaluation criteria valid as from now.
- Check personnel and time schedule.
- Initiate corrections.
- Include supplement to original evaluation contract with the altered conditions.

Phase 3 EVALUATION OF THE CONTENT OF DOCUMENTS AND OBJECT

Stage 1

Installation by the manufacturer of the object to be evaluated. Familiarization of the evaluation team with the object.

Stage 2

Definition of the detailed work plan Who works on which components when and for how long? This is done in several working sessions of the whole team. Through the first review of the documents and familiarization with the object, each evaluator can plan detailed work packages in conjunction with the catalog of criteria and the preliminary work of phase 2, stage 2. First time estimates should be made against the background of the assurance level, aimed at complexity and own experience. Consideration of total time need not be taken into account in this phase. Then review with time coordination in respect to the overall time schedule.

Stage 3

Start of the evaluation according to the work plan. The project manager has assigned to the individual team members the first work packages. Initially they concentrate on gaining the most detailed possible insight into the system during the processing of the work packages. The time for the first progress review meeting will be fixed.

Stage 4

Progress review meeting Here a first status report of the progress of the work is given and if necessary a correction/improvement list drawn up and handed over to the manufacturer. Based on this list and the manufacturer's answers it can be concluded whether there are indications which make the assurance level aimed for or the enforcement of the security requirements doubtful. In the case of serious deficiencies the decision may be taken to break off the evaluation, but at this time this should be considered the absolute exception. The time delays possibly occurring through the corrections/improvements must be integrated in the work plan. Fix date for next progress review meeting.

Stage 5

Evaluation in accordance with work plan.

Here the evaluation enters its critical phase since due to the intensive work on the content of the product and the tests, a number of potential vulnerabilities may be revealed. Internal reviews must specify for each case whether the positive completion of the evaluation is at risk. In the event that this is so, a project progress review should be convened earlier then originally planned. How often the evaluation cycle is initiated according to work plan and project progress review depends on the complexity of the system to be evaluated. The resulting corrections/improvements naturally cause delays in the time schedule, so that for this reason the number of iterations should and must be limited.

Stage 6

Evaluation of the individual results The results of the individual work packages are presented and a suggestion is prepared as to which criteria are enforced at which assurance level.

Phase 4 PRELIMINARY WORK FOR ISSUING THE CERTIFICATE

Stage 1

The individual evaluation results are summarized in an internal document indicating the assurance level awarded for the security requirements to be enforced.

Stage 2

Preparation of an evaluation report for the manufacturer of the product.

This evaluation report need not list all the individual evaluation results. However it should contain remarks as to how and where the manufacturer can improve his product.

Stage 3

Filing of all results, return of the product to be evaluated, where applicable also of hardware and special documents.

Phase 5 PREPARATION OF THE CERTIFICATE

The preparation and issue of the certificate are the responsibility of GISA.

Stage 1

If the evaluation was not conducted by the evaluation authority but by an authorized evaluation office, all the documents drawn up and the results shall be passed on to the evaluation authority.

Stage 2

Review of the evaluation results by the evaluation authority.

Stage 3

Inclusion in the list of evaluated products. Issue of the certificate and presentation to the manufacturer.

Notes on the two terms, list of documents, work plan.

The documents made available by the manufacturer for the evaluation are compiled in a list which is subsequently included in the non-public appendix to the certificate. It is not the aim and purpose of this list to contain as many entries as possible. The manufacturer should not flood the evaluation authority with documentation irrelevant for the evaluation. The list should only contain documents which make statements on the security requirements to be enforced by the system. This naturally relates to all stages of the software development process and the associated documents. Thus if user identification is required in the security requirements, all the documents in which this is addressed shall be included in the list down to the program description of the appropriate module which performs the user identification. So that this list really only contains the relevant documents, a pre-examination of the security requirements is therefore necessary.

In the course of the evaluation the work plan will naturally be changed and refined frequently, but it is the main document which dictates the sequence of the evaluation. The first version is drawn up in phase 2, stage 2. In order to be able to create this document the evaluation team shall be familiar with the general design and structure of the system to be evaluated.

The detailed work plan is then only prepared in phase 3, stage 2. It should take improvement/correction cycles as part of the evaluation to a certain degree into consideration. As a result of the handling of the documents, the discussions with the manufacturer and the initial experience obtained of the system itself, there is a sufficient information baseline available. The work schedule should, only in well-formed exceptional instances, be amended from this point onwards.

7.5 Assessment Steps of an Evaluation

During the evaluation of an IT system the system components to be evaluated are assigned to individual work packages for the evaluators to be rated, in accordance with the IT security criteria catalog. The evaluators have to pay attention to the fact that they do not become bogged down in irrelevant details during the evaluation (e.g. search for implementation errors, which have nothing to do with the evaluation).

In an evaluation, what is checked is whether the security function of the IT system enforces the security requirements.

The processing of a work package is done in several assessment steps. It is an iterative process, i.e. it may be necessary to return to assessment steps which have already been concluded. This can, for instance, be the case if, during the assessment of a security function, dependences to other system components are discovered, so that further documents for this processing are needed. A possible procedure for the processing of a work package is presented here:

Assessment Steps

1. Required documents are compiled.
   (Security requirements, specification of the system components to be processed, description of interfaces, manuals)

2. Overview of the work package, created from the standpoint functionality and system embedding.
   (security requirements, specification)

3. Security requirements concerned to be identified and their consistency tested.
   (Security requirements)

4. Security-relevant functions identified and allocated to the basic security functions.
   (Specification)

===> Current results presented in the evaluation team.

5. Detailed familiarization process with the work package.
   (Specification, description of interfaces, manuals)

6. Mapping of the individual hierarchy levels of the specification (from Q3 down to the source code) to be tested.
   (Specification, if necessary source code)

7. Dependencies to be established to other system components.

(Specification)

8. Separation mechanisms to be identified and assessed.
   (Specification of the separation, description of the interfaces)

9. Mechanisms of the security functions to be identified and assessed.
   (Specification, if necessary source code)

10. Coverage of the security requirements by the security functions to be tested.
    (Security requirement, specification, description of the interfaces)

====> Assessment ratings to be presented in the evaluation team.

11. Documentation of evaluation steps and results.

7.6 The Certificate

The certificate to be issued is the final document of an evaluation. It consists of three parts. Firstly, the certificate itself, which contains the basic statements concerning the evaluated system and is available as a public document. Secondly, an appendix which contains detailed specifications to the evaluated system and is equally available to the public. Thirdly, an appendix 2 which similarly contains details on the evaluated system but is not made public but is only intended for the manufacturer and the evaluation authority. There are amendment lists for both appendices.

The information contained in the three parts is listed in the following.

1. Certificate

   - System name with version and revision level.
   - Hardware configuration on which system was evaluated, with revision level.
   - Assurance level awarded.
   - Class(es) of functionality achieved or description of the security requirements enforced.
   - Version of the catalog of criteria used for the evaluation.

2. Appendix 1 (public)

   - Description of the evaluated software configuration with notes as to whether the certificate also applies to other software configurations.

- Description as to what other hardware components (with revision level) are covered by the certificate or note that no changes to the hardware configuration are allowed.
- Detailed description of the security requirements enforced.
- List of the user documents belonging to the evaluated system.
- Description of the evaluation with notes on critical areas.

List of changes to Appendix 1

- Changed entries relating to points in Appendix 1.

3. Appendix 2 (not public)

- List of modules, functional units or components which were evaluated, with version and revision level.
- List of modules, functional units or components which may not be altered, with version and revision level.
- List of tools which were used in the development process of the system, with version and revision level.
- List of the overall documentation used in the evaluation.
- Brief description of where vulnerabilities in the evaluated system exist.
- Brief comments on where the assurance level of the system could be improved, or note that a higher assurance level cannot be achieved.

List of changes to Appendix 2

- Changed entries relating to points in Appendix 2.
- Other changed entries due to the rules laid down for re-evaluation.

7.7 Consequence for the Manufacturer

If a certificate is issued for a system, two important consequences result for the manufacturer.

1. The rules stated in the chapter on re-evaluation shall be followed, i.e. modifications are subject to strict controls by the evaluation authority. If rule R4 can be applied, the evaluation authority shall at the very least be notified. Otherwise the certificate loses its validity.

2. If the manufacturer must and wishes to make changes to an evaluated system which the evaluation authority cannot or will not accept for any reasons whatsoever, the system shall receive a new version number. The certificate is naturally no longer valid for this system.

8. Developmental Evaluation

The evaluation offices can also carry out developmental evaluations, in addition to the evaluation of a finished product. In a developmental evaluation the product to be evaluated is not a finished product but rather it is still under development or is being further developed during the evaluation. In general, a series of special features are linked to an a developmental evaluation and these differentiate it from the evaluation of a finished product.

The development methodology of the manufacturer if it is appropriate for the assurance level aimed for should be adopted and laid down in the project plan for the sequence of a developmental evaluation. It cannot be expected of the manufacturer that he bases his development process on the evaluation sequence. The schedule for the developmental evaluation will basically be oriented towards the schedule for the product under development.

The documentation to be submitted for an evaluation (e.g. security policy, specification of the security functions, security manuals, etc.) is usually not available at the beginning of a developmental evaluation or is still under development. Thus it is, on the one hand, possible at a very early stage to make the manufacturer aware of errors and gaps in the documentation; on the other hand, the number of amendment cycles will be very large. This is, in particular, in the case of the specification documents a far from negligible problem. Frequent amendment cycles in the specification documents world turn the evaluation into a lengthy process, as already evaluated specification documents would have to be evaluated several times due to design changes. Therefore, for developmental evaluations, it is particularly important to fix a point in time beyond which the documents may only be amended in exceptional cases. Only from this time onwards should the evaluation of the relevant documentation be begun.

The same applies - and in particular at higher assurance levels - for improvement/correction cycles of the product itself. Here again the number of improvement/correction cycles should be so restricted that the sequence and schedule of the evaluation are not placed at risk.

The rating of the manufacturing process is undertaken for a developmental evaluation in the course of the development process. For this, the manufacturer must allow the evaluators to look at all the development and quality assurance steps during the development of the product. At higher assurance levels, a developmental evaluation has more the character of a quality control, as all phases of the development process are monitored by the evaluation team and problems occurring can be taken care directly in the development process.

During a developmental evaluation attention must be paid to the fact that the evaluation team is continuously fully occupied during the evaluation. This can be a problem when the individual phases of the evaluation do not form a continuous chain and there are periods of inactivity for members of the evaluation team or when the development process does not run continuously due to improvement/correction cycles. What should not be permitted is that evaluators participate in several evaluations on different products.

The special features concerning developmental evaluations have to be correspondingly considered in the description of the evaluation process in Chapter 7.

9. Reevaluation

Once a system has been subjected to an evaluation, it is unrealistic to assume that it is therefore free of errors or not liable to any modifications. Since we live in a changing environment, a system will have to satisfy other requirements. These are naturally reflected as modifications to the software.

The question thus arises of how these software modifications to an evaluated system are to be treated. In view of the time and money involved in the evaluation of a major system, it is understandable that one is unwilling to start an evaluation for every modification.

On the other hand it must also be considered that an evaluation and its associated certificate make a statement on the enforcement of a security policy on which the user relies. The evaluation authority is now faced with the problem of having to allow unevaluated modifications on the one hand, while on the other hand a kind of "guarantee" for the product is accepted.

It is clear that there are modifications which require a completely new evaluation. An example for this would be the restructuring of the kernel of an operating system. However in the same way there will be modifications which only involve a partial evaluation and for which the time might only be a matter of hours. An example of this is the elimination of a simple implementation error in an evaluated component.

However there will also be modifications which do not involve any evaluation. When a system is evaluated, those parts which are directly necessary for enforcing the security requirements are evaluated first. These are followed by those components which are not sufficiently separated from the security relevant components and can thus influence the security relevant parts in the event of faulty behaviour. Thus if changes are made to the system which do not affect these parts, there is no need for a reevaluation.

As regards the effort, the assurance level at which the system was evaluated has an effect. It is understandable that a modification to a system evaluated at assurance level Q6 raises quite different questions compared to a modification of a system which has only achieved assurance level Q2. Thus a variety of factors influences the effort caused by a reevaluation.

However what must be expressed quite clearly is the fact that any modifications made by the manufacturer or the operating authority to the executable code declared stable of an evaluated system and which are not reported or only subsequently reported to the evaluation authority invalidate the certificate. As shown in the chapter entitled "Certificate", the list contains those parts which may not be altered after the evaluation since they contribute directly towards enforcing the security requirements. Naturally this does not mean that other system configurations cannot be generated which become necessary because of an altered hardware installation. However if the certificate relates to a specific hardware configuration, then with the changing of the hardware configuration the certificate naturally loses its validity.

The philosophy behind the decision for a reevaluation distinguishes between evaluated and non-evaluated components. These components may be purely software parts, but also other documents, e.g. a specification document. The evaluated software components consist of parts (T1) which are directly necessary to enforce the security requirements. These are identical to the security functions which realize the security requirements, and of additional parts (T2), which are used by the security functions as services, e.g. a sorting routine or a search routine. In addition we have those software components (T3) which could not be separated enough from the security functions due to the design of the system.

At the higher assurance levels the reevaluation is also influenced by the tools (T4) used to develop the system. For instance if the code generator of a compiler which was used for a system evaluated at assurance level Q6 is altered, a new version of the system translated with this compiler is not automatically granted assurance level Q6.

The following rules of thumb for reevaluation can be derived from this:

R1) Modifications and/or extensions to T1 generally require a reevaluation by the evaluation authority.

R2) Modifications and/or extensions to T2 and T3 are sent to the evaluation authority together with the documents necessary for the assurance level. The evaluation authority then decides whether a reevaluation will be necessary or not.

R3) Modification and/or extensions to T4 are sent to the evaluation authority together with explanatory documents. The manufacturer is informed whether there are any objections to the use of the modified parts T4.

R4) Modifications and/or extensions to non-evaluated parts have no influence on the certificate.

A few examples are shown below indicating when and in what direction reevaluation is necessary, and how the above rules are applied.

Example 1:

Elimination of an implementation error in a compiler used for a system evaluated at Q6.

Evaluation: Elimination of the error has no negative effect on the correctness of the machine instructions generated.

Evaluator: Manufacturer, evaluation authority for information with release authority.

Certificate: No impact on the certificate. Entry in list of changes of appendix with reference to new compiler version.

Rule: R3, no reevaluation.

Example 2:

Elimination of an implementation error in the mechanism of a basic security function.

Evaluation: No undesired side effects on other evaluated components. No negative influence on the effectiveness of the mechanism.

Evaluator: Evaluation authority, evaluation, can be performed on the manufacturer's site.

Certificate: No impact on certificate. Entry in the list of changes in the appendix with reference to eliminated error and modified functional units.

Rule: R1, reevaluation necessary, effort dependent on assurance level, complexity and architecture of the system.

Example 3:

Extension of the capabilities of a basic security function with introduction of a new mechanism.

Evaluation: Extension is checked according to the documented assurance level in the certificate taking into account side effects on other evaluated components.

Evaluator: Evaluation office, evaluation can be performed on the manufacturer's site, evaluation authority for information and release authority.

Certificate: New issue with security requirements enforced and assurance level achieved. List of changes contains reference to first certificate. Lists in the appendices of the certificate contain characteristics of the newly evaluated components.

Rule: R1, reevaluation of all affected parts. Effort dependent on assurance level, complexity and architecture of the system. .

Example 4:

Replacement of a mechanism which does not enforce any part of the security requirements by a runtime optimized mechanism with the same interfaces in the privileged part of a system which is evaluated at Q2.

Evaluation: No undesired side effects on evaluated components.

Evaluator: Manufacturer, evaluation authority receives documentation and tests per formed.

Certificate: No impact on the certificate. Entry in the list of changes with reference to new components. The tests by the manufacturer were found to be sufficient.

Rule: R2, no reevaluation only sight test, if manufacturer's documents and tests at assurance level Q2 are appropriate.

Example 5:

Extension of the functionality of non-evaluated parts (e.g. command interpreter) of a system evaluated at Q4.

Evaluation: No influence on the mechanism which separates the evaluated component from non-evaluated components.

Evaluator: Manufacturer, evaluation authority is notified.

Certificate: No impact on the certificate.

Rule: R4, evaluation authority knows that separation mechanism at Q4 is sufficiently strong. Hence faults in the modified parts have no influence on the quality of the security relevant parts.

Regarding the impacts of reevaluation on a certificate, we can distinguish between two cases:

a) The certificate as a whole becomes invalid, a new one is issued.

b) The certificate retains its validity and entries are made in the list of changes in the appendices. The changes are explained there.

10. Evaluation of IT Systems which Contain Already Evaluated Components

It is the goal of the catalog of criteria to cover the largest possible spectrum of IT systems. Thus on the one hand it is possible to evaluate systems developed for very special applications, while on the other hand even complex systems consisting of several components linked together shall be evaluable. It can occur that individual components of such a complex system have already been evaluated.

In order to simplify the evaluation of such systems, the results of the evaluations of the individual components shall be included in the evaluation process. To allow this the sponsor shall provide the evaluation authority with an additional document specifying precisely which security requirements for the overall system are to be enforced by individual components which have already been evaluated, and how the separation of the individual components from one another is assured. The structure, degree of detail and form of this document depends on the assurance level aimed at and should correspond to the presentation of the security policy for the overall system or the presentation of the specification.

The evaluation authority then checks whether the security requirements for the individual components thus derived are a part of the security requirements tested during the evaluation of these individual components and whether the individual component was evaluated at an assurance level which is equal to or better than the assurance level aimed at to for the whole system. If this is the case the individual component need not be considered further within the framework of the evaluation of the overall system. All that has to be checked for the rating in accordance with the assurance level aimed at is whether the component enforces its part of the security requirements of the overall system and cannot be bypassed in any way. A further point to be examined is whether the individual component in the overall system is separated with sufficient assurance from the not evaluated parts of the system.

Example:
Let us assume a network consisting of a several computers evaluated at assurance level Q2, trusted interface unit at assurance level Q6 and file servers of assurance level Q4.

The security requirements are the following:

Identification and authentication of users. It must be possible to combine data in files. It must be possible to assign an attribute to users and files, the values form a hierarchy. This attribute shall be used to allow the system to enforce a security policy consistent with the axioms of the Bell-La Padula model.

Let the security requirements be distributed among the individual components as follows:

Identification and authentication are performed by the individual computers. The evaluation of the labels and enforcement of the Bell-La Padula axioms is enforced by the trusted interface unit. The file administration and the storage of labels and authentication information is performed by the file servers.

Hence the overall system can achieve assurance level Q2 at the most, since part of the security requirements are enforced by a component evaluated at Q2. In order to achieve a higher assurance level for the system as a whole, either the individual component must be evaluated at a higher level in a reevaluation or the security requirements enforced by this component (identification and authentication of users) must be realized by a component with a higher evaluation rating (e.g. trusted interface unit). However the additional parts of the security requirements now to be enforced by this component must be a subset of the security requirements to be enforced by this component during its evaluation. A higher rating of the whole system is then possible if the evaluation reveals that the necessary security functions of the individual components cannot be bypassed or invalidated in the overall system.

11. Tools and MethodsGeneral Remarks

As illustrated for the higher levels, the requirements made on the software development process and the presentation of the security policy and the specification increase from level to level. The large number of methods and associated tools needed will eventually prove to be a problem to the evaluation authority and other evaluation units. In the long run it is not possible for the authority to train the staff at its own cost to enable them to use the tools and master the methodology used in each evaluation.

A solution to this problem is also in the interest of the manufacturers, since at higher assurance levels a certain familiarity is necessary and this cannot be achieved within a short time. Thus the evaluations would drag on inordinately. There is also the risk that due to lack of familiarity systems might be assessed wrongly. This is not in the interest of the authority. It will therefore be forced to maintain a list of tools and methods which it authorizes for the higher assurance levels.

Model Structure of the List of Methods and Tools

- Methods or name of tool.

- Brief description and area of application.

- If standardized, standard number otherwise manufacturer, distributor.

- Versions authorized.

- Use admitted for the following assurance levels (list).

- Will not be accepted any more in the future as of (date).

- Will be accepted until (date) at least.

- Other notes.

12. Mapping on Other Catalogs of Criteria

It is known that other nations and bodies are also considering whether to develop an own catalog of criteria to evaluate IT systems or whether they should adopt the criteria of the US Department of Defense. For instance, draft catalogs of criteria have already been published in Great Britain and Canada. Therefore the mapping on the TCSEC is presented here.

During the period 1980-1983 the US Department of Defense funded investigations aimed at creating a catalog of criteria to evaluate the trustworthiness of computer systems. This document, called the "Trusted Computer System Evaluation Criteria", better known under the name "Orange Book", first appeared on 15.08.1983. In it four groups (D, C, B and A) with altogether seven classes (D, C1, C2, B1, B2, B2 and A1) were defined. For each of these classes criteria to be fulfilled for the four areas Security Policy, Accountability, Assurance and Documentation are established. The criteria for these four areas become more detailed from class to class, so that the seven classes form a hierarchy whereby D is the lowest class and A1 represents the highest class. Functionality and assurance are thus coupled. Although this leads to a clear number of classes, it has become apparent that there is a number of relevant systems which do not fit into this categorization and thus cannot be evaluated in accordance with these criteria. Therefore an approach was selected for the German catalog of criteria in which the criteria of functionality and assurance are separated. Thus for the appropriate class of the Orange Book on the one hand the equivalent class of functionality and on the other hand the equivalent assurance level must be identified. The first classes of functionality of the German catalog of criteria are formulated in such a way that they cover to the best knowledge of the authors the functionality required by the Orange Book in its classes C1 to A1. The assurance levels were developed completely independently of the Orange Book and contain a number of criteria which affect the quality of a software product, but which are not contained in the Orange Book. From this follows that a system which was evaluated in accordance with the German catalog of criteria should certainly fulfil the equivalent class of the Orange Book for the same functionality and appropriate assurance (see Table 1), but also that the converse conclusion is not possible. A system evaluated on the basis of the Orange Book always satisfies a corresponding class of functionality, if necessary one yet to be defined, but does not automatically fulfil an assurance level. Additional criteria must be satisfied for this. At the lower assurance levels in particular this should not lead to major problems if the manufacturer is cooperative. How the mapping in both directions will look in detail still has to be coordinates with the Americans.

In the Orange Book, the security-relevant parts of a system are called the "Trusted Computing Base (TCB)". The TCB comprises all the system components which are responsible for the enforcement of the security policy and the protection of the objects in the system. It consists of the security-relevant software, hardware and firmware components. In the interests of comprehensibility and maintainability, the TCB has to be as small and understandable as possible.

In order to fulfil these requirements, the Orange Book demands the realisation of the TCB according to the "reference monitor concept". The reference monitor validates each user access to programs or data in the system on the basis of a user access list. The reference monitor has to fulfil three design requirements:

1. The reference monitor must be protected against unauthorized access (tamperproof).
2. The reference monitor may not be bypassed (always invoked).
3. The reference must function correctly.

As the IT catalog of criteria are formulated more generally in many points than is the Orange Book, the terms "Trusted Computing Base" and "Reference Monitor" are not used in the IT catalog of criteria. However a mapping between the two catalogs of criteria can be done here as well.

The TCB is formed by those system components in the IT catalog of criteria

1. which perform the security functions.
2. which perform the necessary system services for the security functions.
3. which are not adequately separated from 1 and 2.
4. which realise separation mechanisms.

The term reference monitor was not used in the IT catalog of criteria because, strictly speaking, it only covers the basic security functions administration of rights and verification of rights.

The requirement in the Orange Book for a reference monitor is covered in the IT catalog of criteria by the security functions of the basic functions administration and verification of rights and the strength of the mechanisms with which these security functions are realised. The reference monitor characteristics are, by contrast, very much required. The requirement in the Orange Book "small enough to be analysed" is dependent in the IT catalog of criteria on the assurance level aimed for on the system structure and the complexity of the IT system. In order to fulfil the requirements of the higher assurance levels, the system components shall be so designed and implemented that their correct functionality and their continuous invocation can be tested and analysed.

German Catalog of Criteria		Orange Book Class
Q0	--->	D
F1, Q2	--->	C1
F2, Q2	--->	C2
F3, Q3	-->	B1
F4, Q4	--->	B2
F5, Q5	--->	B3
F5, Q6	--->	A1
Q7	--->	Beyond A1

Table 1: Mapping Between Two Catalogs of Criteria

Explanations:

As can be seen from the first entry, classification at Q0 is independent of the functionality and this corresponds exactly to the philosophy of the Orange Book for Class D. All systems which have not achieved any higher assurance level belong to this class. The further entries indicate a characteristic feature of the Orange Book, i.e. that there is no difference in assurance in the systems classified under C1 and C2, i.e. the criteria for assurance are the same. In contrast to this the differences between a system classified under B3 or A1 is largely in the area of assurance. The functionality is the same. The level Q1 defined in the German catalog of criteria is intended for a relatively simple and short-term evaluation of a system. The manufacturer is hardly involved in the evaluation. A potential user of a system receives a statement about the minimal quality of the system very quickly. Due to the sequence of this evaluation it is also possible to indicate whether evaluation at a higher assurance level appears feasible. All this is intended to relieve the evaluation authority, since evaluation at a higher assurance level entails considerable organizational preparations which should not be carried out in vain. Assurance level Q7 describes criteria which do not exist in the Orange Book. All that is mentioned there is a class to be defined in the future which is currently summarized as "Beyond A1".

Glossary

Accountability:

Auditing of the exercise or attempted exercise of rights, in particular to be able to prove violations of security policy after the event.

Administration of Rights:

Part of the system which administers the right status between subject and object, e.g. in the form of administration of an access control list.

Assurance:

Measure for the quality with which certain requirements of an IT system are fulfilled. In accordance with DIN 55350: overall system of characteristics and features of a product or an activity which refer to the suitability for the fulfilment of given requirements.

Assurance Levels:

Hierarchical classification as regards the assurance of an IT system. In the evaluation the assurance of an IT system is rated. On the basis of this rating classification at one of assurance levels Q0 to Q7 takes place (see Chapter 6.2 of IT catalog of criteria).

Authentication:

Evidence of the given identity.

Availability:

Probability that data stored and processed on an IT system are accessible at an specified time or that the IT system is in working condition at a specified time. In another context: probability that a system is found to be in working condition at a specific time.

Class of Functionality:

Class which sets specific minimum requirements as regards the functionality of the security functions of an IT system.

Configuration:

Selection of one of the possible constellations (of hardware and software) for an IT system.

Covert Channel:

Communication channel which permits a flow of information in violation of the security policy. The bandwidth of a covert channel is a measure for the possible amount of data transferred per unit in the communications or information technology senses.

Data Communication:

Transport of data, whereby the communication channel is generally not secured by the functions of verification of rights and administration of rights.

Bell-La Padula Model:

Formal, rule-based security model (predominantly for security requirements in the field of confidentiality of data).

Debugger:

Software tool for error search.

Evaluation:

Examination and rating of an IT system on the basis of the catalog of criteria.

Evaluation Authority:

This authority coordinates the evaluation and certifies the evaluated product in accordance with the rating.

Evaluation Manual:

Guidelines for the procedure to be adopted for the evaluation.

Evaluation Report:

Document in which the result of the evaluation of the IT system and the resulting rating divided into individual aspects is recorded.

Formal Evidence:

Strict evidence in the mathematical sense, generally based on predicate logic.

Formal Model:

A formal model consists of quantities of abstract objects with functions and operations defined thereon and for which certain laws (axioms) apply. The basic skeleton for a formal model is e.g. the predicate logic. A model develops from reality by abstraction (and hence simplification) and is accessible to mathematical treatment.

(Formal) Security Policy Model:

(Formal) model for the security policy or parts thereof.

IT System:

Information technology system.

Identification:

Determination of the identity of a subject or an object.

Integrity:

Measure for the non-manipulation and correctness of data.

Design Specification:

A specification which describes the design of an IT system or an IT system component. Typical contents: control flow, data structures, data accessed, input/output formats, algorithms used and interface descriptions.

Basic Functions:

Abstract description of the security policy for an IT system. They can be used to group security requirements together.

Functional Units:

Module, procedure, compilation unit or component of the software of an IT system.

Identification:

Determination of the identity of a subject or an object.

Mechanism:

Description of a procedure (solution) as to how one of more security requirements made of an IT system can be enforced by it.

Object:

Objects play the passive role in the administration of rights or verification of rights, i.e. they are accessed. Example: files, tables of contents, equipment.

Object Reuse:

Treatment of reusable facilities such as e.g. main memory or external storage media in order to prevent any unauthorized flow of information between subject or object which use this facility consecutively.

Operational Behavior:

Measure for the enforcement of the security policy during the operation of an IT system, in particular in exceptional situations such as errors and maintenance. Similarly in other context: measures for the enforcement of all functional requirements during the operation of an IT system, in particular in exceptional situations such as errors and maintenance.

Penetration:

Bypassing the security policy of an IT system.

Penetration Test:

Test aimed at examining security functions for the possibility of penetration.

Role:

A role is a grouping of rights allocated to a subject. Example: the role of the system administrator.

Security Functions:

Functions which realize the security requirements in the IT system.

Security Policy:

A number of requirements and rules which specify how security relevant data is to be handled and processed.

Security Relevant Event:

An event which can cause violation of the security policy.

Side Effect:

Unintended, unspecified secondary effect of a function which under certain circumstances can also cause or permit a violation of the security policy.

Integrity:

Characteristic of data which is equivalent to their correctness with respect to stated criteria.

IT System:

An information technology system.

Risk:

Probability that a threat for this IT system is effective and damage occurs due to the vulnerability of an IT system.

Specification:

A document which describes the requirements made of the system or of parts of the system, the architecture, the behaviour or other characteristics precisely, comprehensibly and comprehensively.

Subject:

Subjects play the active role in the administration or verification rights, i.e. they exercise rights over objects, e.g. persons and processes.

System Administrator:

Role in the administration of rights, generally with exceptional rights.

Threat:

Factor or event which jeopardises the confidentiality, integrity and availability of the data processed and stored on an IT system or the availability of the IT system itself.

Trap Door:

A hidden functionality of the system which permits the security policy to be violated.

Trojan Horse:

A program which apparently performs a useful task but carries out other actions covertly, e.g. using the privileges or rights of the caller to circumvent the security policy.

User:

Person in contact with the IT system and who avails himself of its services and functions.

Verification:

Evidence of the correctness of programs with formal means, e.g. "wp" conditions.

Verification of Rights:

Examination by the system of whether a certain subject is entitled to access the desired object in the intended manner. Verification of rights prevents unauthorized exercise of rights.

Vulnerability:

Weak point in the IT system which can be exploited to bypass or violate the security policy.