UNIVERSITÄT
HILDESHEIM

Universität Hildesheim • Marienburger Platz 22 • W-3200 Hildesheim

Dr. Andreas Pfitzmann
Institut für Informatik
Gebäude Samelsonplatz 1

☎ +49 5121 883735 (office)
☎ +49 5121 83312 (home)
Fax +49 5121 860475
e-mail pfitza@infhil.uucp

Commission of the European Communities
Directorate XIII/F
SOG-IS Secretariat
Rue de la Loi, 200
B - 1049 Brussels
Belgium

29th Nov. 1991

Subj. **ITSEC Comments to V1.2**

Dear Ladies, dear Sirs,

following please find my **Statement of Observations** concerning the Information Technology Security Evaluation Criteria (ITSEC), Provisional Harmonised Criteria, V1.2, 28 June 1991.

On April 17th, 1991, I participated in the ITSEC V1.1 Workshop, and some answers from the representatives of the group of four were all but convincing to me, in particular concerning my general observations 1.1, 2.1, and 6.1. The same is true for ITSEC V1.2 and its accompanying response to key comments raised by reviewers.

Even now, a real high quality review of ITSEC is impossible, since reviewers lack the information of ITSEM and experience with many different functionality classes to be defined. Therefore, it is impossible to decide impartially at the moment whether Christian Jahl's answer to my general observation 2.1, that this incompleteness can be solved by "defining a functionality class using exclusion functions", is appropriate or not.

ITSEC V1.2 has most of the observed deep problems of V1.1 just as V1.1 has most of the observed deep problems of V1.0. Therefore not much prophecy is required to predict that V2 will have most known deep problems if everything goes further as usual. I admit that at the moment, we are lacking worked out alternatives. Therefore, the CEC should sponsor the development of alternative proposals and evaluation experiences for Version 2 of ITSEC.

If there is new material, e.g. a synopsis of all statements of observation, please send it to me. At any time, I am interested to receive answers to my questions and feedback about my comments and/or the future versions of ITSEC and ITSEM.

I appreciate the comments of Kai Rannenberg, Birgit Pfitzmann, Michael Gehrke, Christian Eckert, and Hans H. Brüggemann.

Yours sincerely

A. General Observations

1. Title and Scope

1.1 Version 1.2 of the **ITSEC** is the best and most general we have today. But it **does not, by far, address the scope suggested by its title (and the scope section) and needed in a future information society**. Therefore, either the title (and the scope section) have to be narrowed or much more and much broader work is required.

- The ITSEC Version 1.2 are even less Security Evaluation criteria than Version 1.0. They define a *framework* to formulate Security Evaluation criteria. In a certain sense this is more, in another this is less than bare Security Evaluation criteria. In any case, it does not help to define evaluation criteria in a way that the ITSEC title becomes correct (in this respect) as done in the presentation of Jonathan Wood on the ITSEC V1.1 Workshop. The TCSEC [Trusted Computer Systems Evaluation Criteria, DOD 5200.28-STD] are with us for many years and there is absolutely no need to use "Evaluation Criteria" with another meaning than there. (Side remark: The argument in [ITSEC V1.1 Revision: Addressing the Main Issues, p.6, §4.1] does not convince: Its result would be either to change "criteria" to "framework" or to change "Security" to "Security testing". But neither is done.)
- The ITSEC do not address *decentrally managed* IT-Systems, i.e. connected IT-Systems with multiple administrations with potentially conflicting interests. I admit that this is a difficult problem but ignoring a very urgent problem in the scope suggested by the title (and the scope section) is very misleading and no good service to society. Reference to non-repudiation under the heading Data Exchange is by far not enough and raises problems of systematics, see 2.1 below.

An appropriate title for Version 1.2 of ITSEC might be: *A Framework of Security Evaluation Criteria for hierarchically managed Information Technology Systems*

1.2 The ITSEC do not cover the **problem of different kinds of potential attackers**, i.e. not only users, but also designers, manufacturers, operators; designers, manufacturers, and operators of the design tools; and so forth. Additionally, in real life systems, a difference has to be made between potential attackers of different parts of the system (or product).

Especially risks caused by the operator of the system are not considered or are not considered any longer.

An example is §E4.35 of V1.2 or V1.1 in comparison with the associated § in V1.0 (p. 62). V1.0 covers the aspect of "unrecognized loss of functionality of the security functions" caused by errors of the system operator. In V1.2 and V1.1 this aspect is left out.

1.3 Definition of confidentiality and integrity

I propose the following definitions (changes in italics):

confidentiality	prevention of the unauthorized disclosure of information*, or, if possible, avoidance of unnecessary information*.
integrity	prevention of *undetected* unauthorized modification of information.

Justifications:

The best (and, at least against strong attackers in today's systems, only) way to keep information confidential in a provable way is to avoid that it can be gathered. Surprisingly, in theory nearly any application can be realized that way and in practice many essential applications can, e.g. communication networks, payment systems, authorization systems, value exchange systems, cf. [David Chaum, Security without Identification: Transaction Systems to make Big Brother Obsolete; Communications of the ACM 28/10 (1985) 1030-1044], [Andreas Pfitzmann, Birgit Pfitzmann, Michael Waidner: ISDN-MIXes – Untraceable Communication with very small Bandwidth Overhead; Proc. Kommunikation in verteilten Systemen, Feb. 1991 Mannheim, Informatik-Fachberichte 267, Springer-Verlag, Heidelberg 1991, 451-463], and [Holger Bürk, Andreas Pfitzmann: Value Exchange Systems Enabling Security and Unobservability; Computers & Security 9/8 (1990) 715-721].

In distributed systems, you cannot prevent unauthorized modification of data, e.g. in transit on a network, but you can detect unauthorized modification with cryptographical means. Additionally, the proposed change of the definition of integrity provides for a cleaner separation of integrity and availability. A third justification is that with the proposed change of the definition of integrity, integrity corresponds to the well established notion of *partial correctness* in program verification, and integrity and availability together correspond to the well established notion *total correctness*.

2. Functionality

2.1 The classes (Generic Headings) for the security enforcing functions are incomplete and unsystematic.

As noted already in my statement of observations concerning ITSEC V1.0, for some services, essential duals of the given classes are missing. E.g. for some services, *Identification* and *Authentication* are wanted, for other services, you need the corresponding duals *Anonymity* and *Pseudonymity*, cf. [David Chaum, Security without Identification: Transaction Systems to make Big Brother Obsolete; Communications of the ACM 28/10 (1985) 1030-1044]. The same is true concerning *Audit* and its dual *Freeness from observability in "private" domains.* Christian Jahl's argument given at the ITSEC V1.1 Workshop that the group of four left out these duals because they feared confusion is more an argument for the contrary: If these duals are essential (which was unanimous consent on the ITSEC V1.1 Workshop) the problem has to be treated by the readers of the ITSEC whose confusion Christian Jahl fears. And their task will surely be more difficult (and their "confusion" greater) if the ITSEC do not introduce and define these duals!

Data Exchange is no security enforcing function at the same level of abstraction as the other seven function classes given, e.g. there is no class Data Storage. Assigning to that eighth function class such important aspects as non-repudiation makes this unsystematic approach even worse.

2.2 The 10 example functionality classes give very poor guidance.

The possibility to define additional functionality classes does not solve the following problem: Without guidance by the ITSEC themselves, users and customers will not be able to specify their security needs. Therefore the 10 example functionality classes give the false impression every relevant security problem is covered.

2.3 Not only in the Chapters 0-6 of ITSEC, but even in the proposed functionality classes, there are no **limits** imposed on the **bandwidth of covert channels**. Compared with the Orange book and [Criteria for the Evaluation of Trustworthiness of Information Technology (IT) Systems, ISBN 3-88784-200-6; German Information Security Agency, 1989], this is a large step backward. The highest security class has to restrict the bandwidth of all covert channels together below $4 \cdot 10^{-12}$ bit/s.

Justification: Imagine we have a file of people including the attribute "AIDS, yes or no". This bit of information has to be kept secret for the lifetime of the patient, e.g. 80 years. It might be acceptable that the file leaks 0.01 bit during this timespan. This yields an acceptable bandwidth of 0.01/80 bit/year = $4 \cdot 10^{-12}$ bit/s.

The very least is that functionality classes dealing with higher assurance of confidentiality require that an upper bound for the bandwidth of covert channels is given. It is then for the accreditor or procurer of a system or product to decide, whether the possible bandwidth is acceptable or not. If the ITSEC do not require to evaluate this bandwidth for e.g. products, many procurers will have to do that, which clearly is a multiplication of effort.

2.4 The incomplete classification of functions and functionality hinders the adequate classification and certification of products and systems which guarantee Anonymity, Pseudonymity and Freeness from observability.

In their current state the ITSEC do not give any help for evaluation and certification of systems which work without the need/enforcement of any gathering of unnecessary person-related information.

Examples of such systems are referenced in 1.3 and known under the following buzzwords:

untraceable communication = networks without user observability;

untraceable payment = digital payment systems without user observability;

authorization without identification;

value exchange systems without user observability.

3. Assurance – Effectiveness

3.1 The discrimination between strengths of mechanisms in only three classes (basic, medium or high) is very poor and not adequate.

There must be more classes, see e.g. [Criteria for the Evaluation of Trustworthiness of Information Technology (IT) Systems, ISBN 3-88784-200-6; German Information Security Agency, 1989, p. 15f] for more classes and good definitions. But even there, there should be at least an additional class I call "unbreakable". You get it from the class of "virtually unbreakable" by omitting "according to the present state of the art", i.e. making the definition time-independent. Examples of members of the class unbreakable would be the one-time pad, cf. [Claude E. Shannon, Communication Theory of Secrecy Systems; The Bell System Technical Journal 28/4 (1949) 656-715], or information-theoretic authentication codes, cf. [Gustavus J. Simmons, A Survey of Information Authentication; Proceedings of the IEEE 76/5 (1988) 603-620].

Having a highest rating which is just "beyond normal practicality" is ridiculous! "Beyond normal practicality" is a trivial condition and only suited to characterize *basic* and not at all *high*.

Subjectivity of the rating is no argument for very few classes. Since if you have only very few classes, you introduce relatively great rounding errors, which is even worse. So define many classes, and if there is subjectivity of the rating, then give a class as expectation and two further classes as an upper and lower limit. I think confidence intervals have a good tradition in engineering!

3.2 The rating of cryptographic mechanisms has to be international and as objective and reproducible as possible. This requires that all mechanisms be published in any detail. Any secret mechanisms are not suitable for certified secure systems.

(Side remark: The argument in [ITSEC V1.1 Revision: Addressing the Main Issues, p.7, §5.2] does not convince at all: If you apply the argument to the ITSEC themselves the argument shows that ITSEC is unnecessary, because everything can be handled as "a mutual recognition issue".)

4. Assurance – Correctness

4.1 As in [Criteria for the Evaluation of Trustworthiness of Information Technology (IT) Systems, ISBN 3-88784-200-6; German Information Security Agency, 1989, p. 53], it should be stated clearly that **evaluation levels beyond E6 are possible, very desirable, and might be defined in the future**.

One reason for this is that there are transitive Trojan Horses, cf. the Turing award lecture by [Ken Thompson, Reflections on Trusting Trust; Communications of the ACM 27/8 (1984) 761-763]. Examples for evaluation levels beyond E6 are

Level E7: Verified design history of all tools used to develop the TOE.

Level E8: Verified design history of all tools used to develop tools used to develop the TOE.

and so forth.

The ultimate level would require that all (recursive definition!) used tools have verified design, i.e. you need some form of secure bootstrap in generating these tools.

(Side remark: The argument in [ITSEC V1.1 Revision: Addressing the Main Issues, p.4, §1.5] does not convince at all: Criteria should define or at least mention possible targets and not what the criteria makers believe to be "the state of the art" which is more a matter of expense than "art" and even if it would be "art", hopefully the "state of the art" emerges faster than (hopefully stable) criteria. Not mentioning what some users of ITSEC might desire, because it is deemed not to be "state of the art", is very dangerous because it gives them a false impression of security.)

4.2 If a **formal security policy for *all* aspects of security** has to exist for the levels **E4 and above** as suggested in §E4.2, in my opinion, this would be a **major (at least: short term) weakness of ITSEC**. Since then, at the moment, only 3 evaluation levels are possible for nearly all relevant TOEs. If the requirement is simply unprecise in formulation, I would assess this general observation as a minor specific observation.

5. Post-Evaluation Problems

Evaluation and Certification of products and systems do not stay valid forever and evaluated products and systems are going to be connected.

5.1 The re-rating and re-evaluation has to be harmonized internationally. Reference to national certification bodies or a statement "beyond the scope" (p. 13) is a bad excuse for not addressing that issue.

5.2 What shall happen if someone discovers (and possibly publishes) a flaw allowing a break of security for a certified TOE? **Does the certificate become invalid by discovery of a security flaw?** How are all users of the TOE notified of this?

5.3 Rating of TOEs consisting of evaluated components. What functionality class and evaluation level should be assigned to, e.g., systems consisting of evaluated products. Is in this case a complete, partial or no re-evaluation required?

6. Development and Discussion of the ITSEC

The organization of ITSEC Development and Discussion must be improved.

6.1 For public acceptability of ITSEC and a constructive scientific discussion, it is not enough to publish the criteria (or more precise: their framework) themselves. **It is necessary to publish a detailed rationale**, i.e. all options and suggestions, the reasons for the selection of options, etc. On a scientific conference on dependable computer systems organized by GI (Gesellschaft für Informatik e.V., the leading German scientific organization in the area of informatics) in March 1991 at Darmstadt, this has been asked for unanimously. "ITSEC Revision: Addressing the Main Issues – A response to the key comments raised by reviewers" is a first step. Others should follow very soon.

6.2 The comparison between V1.1 and V1.2 gives the impression that **critical points are neither discussed nor clarified but left out** in the course of the document's development. Examples are:

§4.24 in V1.1 was criticized, because it did not cover the point that configuration control for tools is important to ensure that insecure functionality is not added. As a result in V1.2 the aspect of additional insecure functionality vanished from §4.24.

§E6.5 and §E6.11 in V1.1 were criticized as the vulnerability analysis mentioned there covered the circumvention of the security of a TOE only in respect to users and left out the risks caused by operators and designers. As a result in V1.2 the vulnerability analysis is not mentioned any longer.

7. Special Problems of IT

Special characteristics of information technology cause problems, which do not exist in other technologies. Therefore, the following is required:

7.1 "**Source code** or hardware drawings for all security relevant components" even **for the levels E1 and E2.**

7.2 "Evidence of **retests** after the discovery and correction of errors relevant to security is obligatory to demonstrate that the errors have been eliminated and no new errors have been introduced." even **for the levels E1 and E2.**

7.3 "**Description of all implementation languages** used" **for the levels E1 and E2.** Given the design complexity of today's IT-systems, black box testing is a hopeless task.

7.4 **Control of the development tools**: The subversion of tools is a special risk of IT compared to other technologies. It is not covered in the ITSEC and should at least be mentioned in the definition of "Developer Security" (p. 113, §6.27) and "Development Environment" (p. 113, §6.28).

B. Specific Observations

pages 24ff:

When you think a bit deeper about the Generic Headings, you will find that some duals are missing. E.g. for some services, *Identification* and *Authentication* are wanted, for other services, you need the corresponding duals *Anonymity* and *Pseudonymity*. The same is true concerning *Audit* and its dual *Freeness from observability in "private" domains.*

It is impossible to decide impartially at the moment whether Christian Jahl's answer to my observation 2.1, that this incompleteness can be solved by "defining a functionality class using exclusion functions", is appropriate or not. At the very least I think that the explanation given in the ITSEC V1.2 is not sufficient to make evaluators, procurers and users aware of the issue to define reasonable models of attackers which have to include insiders, be it they added malicious code during the design or be it the system operator cannot be trusted completely.

page 25 §2.34:

Why did "token" and "physical characteristic" disappear? Please note that the ITSEC are called information *technology* security evaluation criteria.

page 25 §2.39:

If there are "rights", are there "duties" as well?

page 26:

Accountability and *Audit* seem to be more narrow than *Access Control* and the missing *Flow Control* and the missing *Inference Control*. (Please note that the criteria shall, as the title suggests, not be restricted to the evaluation of operating systems.)

page 28 §2.56:

In the ISO Security Architecture, some functions are missing as well, e.g. provable freeness from observability.

page 36 §3.5:

What is the difference between a "critical security mechanism" and a non-critical security mechanism and how does the notion of "security weakness" contribute to this difference? For a natural definition of security weakness, isn't it necessary even to assess only indirectly critical security mechanisms for their ability to withstand direct attack? Since their failure could create weaknesses for other indirectly critical and finally "critical" security mechanisms.

Putting the question the other way round: Since there seem to be no "non-critical" security mechanisms (if there are, they are superfluous functionality and must therefore be excluded from the TOE), why not omit "critical" alltogether?

page 36 §3.5-3.8:

Why not more ratings, e.g. [Criteria for the Evaluation of Trustworthiness of Information Technology (IT) Systems, ISBN 3-88784-200-6; German Information Security Agency, 1989, p. 15f] for more classes and good definitions. But even there, there should be at least an additional class I call "unbreakable". You get it from the class of "virtually unbreakable" by omitting "according to the present state of the art", i.e. making the definition time-independent. Examples of members of the class unbreakable would be the one-time pad, cf. [Claude E. Shannon, Communication Theory of Secrecy Systems; The Bell System Technical Journal 28/4 (1949) 656-715], or information-theoretic authentication codes, cf. [Gustavus J. Simmons, A Survey of Information Authentication; Proceedings of the IEEE 76/5 (1988) 603-620].

Having a highest rating which is just "beyond normal practicality" is ridiculous! "beyond normal practicality" is a trivial condition and only suited to characterize *basic* and not at all *high*.

Subjectivity of the rating is no argument for very few classes. Since if you have only very few classes, you introduce relatively great rounding errors, which is even worse. So define many classes, and if there is subjectivity of the rating, then give a class as expectation and two further classes as an upper and lower limit. I think, confidence intervals have a good tradition in engineering!

page 37 §3.10:

"If covert channels are identified which exceed these bandwidths, ..." then there is nothing to determine by the evaluator: the evaluation has to fail!

page 37 §3.11:

"Construction - Phase 1 - Requirements" in Chapter 4 is referenced to contain an explanation of the differences between "system" and "product". Neither does the referenced paragraph contain this differences nor are the differences set out in surrounding paragraphs.

page 38 §3.16:

The evaluator checks the suitability analysis only in a formal way and with no respect to its content. This is not enough.

page 39 §3.23:

"in the case of cryptographic mechanisms,.... from the appropriate national body." This formulation gives rise to severe doubts: The rating of cryptographic mechanisms has to be *international* and as *objective* and *reproducible* as possible. This requires all mechanisms to be published in any detail. Any secret mechanisms are not suitable for certified secure systems.

page 41, §3.28:

What are the "other known construction vulnerabilities" the evaluator has to take into account in his independent vulnerability analysis? Is there a common knowledge of construction vulnerabilities and where is it defined?

pages 9, 45ff:

There may be higher levels, e.g.

Level 7: Verified design history of all tools used to develop the TOE. Bandwidth of all covert channels together below 10^{-3} bit/s.

Level 8: Verified design history of all tools used to develop tools used to develop the TOE. Bandwidth of all covert channels together below 10^{-6} bit/s.

and so forth.

The ultimate level would require that all (recursive definition !) used tools have verified design, i.e. you need some form of secure bootstrap in generating these tools, and that there are no covert channels.

If covert channels shall be treated separately, this can easily be done and does not change the fact that the evaluation levels form an open series towards an ultimate level.

page 50 §4.16:

Does the "Quality Assurance Programme" encompass the tools used to design the TOE as well? If so, to what rigor? If not, where is the resulting security leak (transitive Trojan Horses, cf. Turing award lecture by [Ken Thompson, Communications of the ACM 27/8 (1984) 761-763]) stated?

page 51 §4.18:

Last sentence may hinder technical progress.

page 52 §4.21:

"... in a way that correctly preserves the intent of the top level description ..." How shall this be checked? Should this informal notion "intent" address even risks which are not mentioned in the text explicitly, e.g. Trojan Horses? (Not mentioning Trojan Horses explicitly is a deficit, since it suggests a degree of security which cannot be achieved today.)

page 52 §4.22:

The second half is wrong in two respects:

Firstly, if a test covers all possible combinations of input values and internal states of the system (exhaustive testing of a *finite* state machine), it can show the absence of errors.

Secondly, it is not enough to show the absence of *errors*. Even the absence of *deliberate subversive functionality* has to be shown.

As a conclusion, "never the absence of errors" should be replaced by "and only in special cases their absence".

page 54 §4.31:

Can this be achieved without tamper-proof boxes? How is their tamper-proofness to be validated, which has explicitly been excluded from the scope of ITSEC, cf. p. 7 §1.3?

page 63 §E2.5, and §§E*n*.5:

"enforcing" should probably be changed to "enforcing and relevant"

page 63 §E2.6, and §§E*n*.6:

"enforcing" must be changed to "enforcing and relevant"

page 63 §E2.7, and §§E*n*.7:

"is valid." How shall this be achieved concerning covert channels?

page 65 §E.2.15:

Since there are transitive Trojan Horses, the demand of the last sentence cannot be fulfilled unless configuration control is extended to the creation of all tools used to create either the TOE or any tool used to create these tools (recursive definition).

page 68 §E2.32, and §§E*n*.32:

"national certification body." This formulation gives rise to severe doubts: The rating of cryptographic mechanisms has to be *international* and as *objective* and *reproducible* as possible. This requires that all mechanisms be published in any detail. Any secret mechanisms are not suitable for certified secure systems.

page 79 §E4.2:

Does the requirement of a "formal model of security policy" for Level E4 and upwards reduce the range of possible evaluation levels for a lot of products to (the lower) three only at least in the near future?

If indeed a formal security policy for *all* aspects of security has to exist it is in my opinion a major (at least: short term) weakness of ITSEC because then, at the moment, only 3 evaluation levels are possible for nearly all relevant TOEs. Other references where this issue is treated are page 46 §4.8 and page 114 §6.37.

page 83 §E4.17:

"Use the developers tools to rebuild selected parts of the TOE and compare with the submitted version of the TOE." This is necessary to find Trojan Horses and should therefore be applied recursively, i.e. rebuild the developers' tools etc.

What if the TOE is an IC? Has the evaluator to produce ICs and to compare ICs? Does he only have to produce the masks and compare them, etc.?

page 101 §E6.13:

Sponsor supplied tools are not sufficient, use independent developed tools.

page 103 §E6.17:

In the highest E-level, not only the TOE should be rebuilt using the developers tools and checked against the submitted version, but the same should be done concerning all tools used to develop the TOE, the tools to develop these tools, and so forth.

pages 111ff:

I would prefer to have a glossary including *all* technical terms. As the second best possibility, provide two glossaries, where the second gives the generally accepted meaning.

page 115 §6.57:

Replace the second word "the" by "a".

page 134 §A.55:

Timing channels as well and not only storage channels?

page 137f §A.73:

Why no bandwidth limit given?

page 148 §A.107:

How shall *end-to-end* encryption ensure confidentiality regarding the *recipient* (nota bene: not only the contents of messages!) over large sections of the communication channel? If it does: Why not confidentiality regarding the *sender* as well?