Proc. 4th RADC Workshop on Multilevel Database Security, Rhode Island, April 1991, pages 122-126.

EVALUATION BY PARTS OF TRUSTED DATABASE MANAGEMENT SYSTEMS

Ravi S. Sandhu

Center for Secure Information Systems
&
Department of Information and Software Systems Engineering
George Mason University, Fairfax, VA 22030-4444

1 INTRODUCTION

A database management system (DBMS) is a superb tool for building effective information systems. The widespread use of DBMS's across the board, from stand-alone personal computers at one end to heterogeneous networked mainframes at the other, is ample testimony to the success and acceptance of this technology. It is therefore no surprise that there is significant interest in trusted DBMS's on the part of users, vendors and evaluators.

The vast majority of DBMS's are hosted on top of some general-purpose Operating System (OS). The open systems thrust which is driving the industry will increasingly lead to situations where the DBMS, OS and possibly hardware are supplied by different vendors. This presents a significant challenge to vendors and evaluators in the development and rating of products.

Over the past few years, the security community has spent considerable effort in examining how the evaluation of a trusted DBMS can be factored out and separated from the evaluation of the underlying trusted OS. This effort has culminated in the recently published Trusted Database Interpretation (TDI) [1] of the Trusted Computer Security Evaluation Criteria (popularly known as the Orange Book) [2].

The authors of the TDI have consciously taken a generic (i.e., non-DBMS specific) and conservative approach to the question of evaluation by parts. This is perhaps appropriate in the first cut at this problem. However, an abstract and generic treatment leads to worst-case scenarios and pessimistic conclusions.

The authors of the TDI have also taken the approach that the Orange Book is a fixed parameter in this exercise. There has been no debate on whether or not the Orange Book could be strengthened or augmented in some way, so as to facilitate evaluation of parts.

In this note we propose some ideas on how the pessimistic conclusions of the TDI can be relaxed. We also speculate on some ways in which the Orange Book might be augmented to make evaluation by parts easier. There are no definite conclusions reached in this note. Our objective is to point out some avenues for fruitful research in the theory and practise of evaluation by parts.

2 EVALUATION BY PARTS

The writers of the TDI have deliberately chosen to take an abstract and (mostly) non-DBMS specific approach to the generic question of evaluation by parts. For instance the TDI states quite explicitly that:

> "The approach taken in this document is to address the issues of evaluating systems built of parts in a way that is independent of the field of trusted database management. This conscious attitude of generality is intended to make clear the distinction between the larger system-of-parts issues and the more specific DBMS issues."

While there are many merits to this approach, the general abstract setting inevitably leads to very conservative and cautious conclusions regarding the efficacy of evaluation by parts.

In particular the following pessimistic conclusion has dismayed many vendors, evaluators and researchers.*

> "This case also concerns a TCB that consists of two candidate TCB subsets, C and D. C is the more primitive subset. That is, D uses the abstractions provided by C .... Additionally, D is trusted with respect to C. That is, some of the C-subjects which make up TCB subset D execute as trusted processes of C. ... This case can be viewed as a special case of a previously evaluated TCB which has been altered. ... Although this case may appear, intuitively, to be different from that of arbitrary alteration of a previously evaluated TCB, the example demonstrates that such an approach makes it impossible to perform an evaluation by parts."

In this case the TDI is clearly treating the addition of a trusted process as an "arbitrary alteration of a previously evaluated TCB." Hence the "impossibility" of evaluation by parts.

*This conclusion was cast in different words in drafts leading to the eventual publication of the TDI as follows: "Thus, a TCB that contains an unconstrained TCB subset will be subject to the full gamut of penetration testing and covert channel analysis appropriate to the target evaluation class for the entire TCB, including any previously or separately evaluated TCB subsets. In the case of TCB subsets less primitive than an unconstrained TCB subset, only local evaluation activities can be done separately." Operationally, the net effect of the revised statement is essentially the same as this earlier formulation.

The reality of system building and software engineering is, however, very different from this hypothetical worst case scenario. Trusted subjects are used to do very specific tasks and require very specific exemptions from the security policy of the underlying TCB.

It is very important for the security community to examine the extent to which the conservative conclusions of the TDI can be relaxed in the light of specific trusted DBMS and trusted OS architectures? We are motivated by the following intuition: the extent to which the underlying trusted OS needs reevaluation should correlate with the degree of match between DBMS and OS architectures. We must go beyond the generic abstract perspective of the TDI to consideration of:

- concrete and realistic trusted DBMS and OS architectures, and
- allocation of overall security policy to individual DBMS and OS TCB subsets.

It is our conjecture that:

- the extent to which the underlying trusted OS needs reevaluation should correlate with the degree of match between DBMS needs and OS services, and
- the extent to which the underlying trusted OS needs reevaluation should correlate with the allocation of the overall security policy to individual DBMS and OS TCB subsets.

3 EXAMPLES

It is worth considering some examples to illustrate why the DBMS architecture and OS architecture are both relevant to this question.

1. Our first example shows the relevance of the OS architecture, specifically concerning the granularity of privilege that can be assigned to a trusted subject. At one extreme let OS-A provide only a binary privilege (such as super-user) for this purpose so that granting the privilege removes all constraints from the trusted subject, i.e., OS data structures are completely exposed to trusted subjects. At the other extreme let OS-B have extremely fine grained privileges to the extent that exemption from the $\star$-property can be granted with respect to individual files. It is intuitively obvious that an unconstrained TCB subset built on top of OS-A should require a greater degree of global penetration testing than one built on top of OS-B with one very specific exemption from the $\star$-property.

2. Next consider the relevance of the DBMS architecture. At one extreme consider DBMS-A which runs as a single trusted subject with respect to the OS. By

definition such a DBMS has the ability to leak its highest sensitivity information to its lowest clearance subjects at essentially instantaneous speed. At the other extreme let DBMS-B run as a collection of untrusted processes, one at each sensitivity level, except for a single trusted process which is used to synchronize the two-phase commit of transactions. It intuitively appears that DBMS-A should require greater degree of global covert-channel analysis than DBMS-B.

3. Finally consider the relevance of the combination of DBMS and OS architectures. Our intuition suggests that the 4 combinations resulting from coupling the 2 OS's and 2 DBMS's outlined above can be ranked as follows in decreasing order of global analysis required.

   (a) DBMS-A, OS-A

   (b) DBMS-B, OS-A

   (c) DBMS-A, OS-B

   (d) DBMS-B, OS-B

   One would expect an order of magnitude difference in effort required at the two extreme ends of this scale.

The challenge is to make a systematic analysis of various architectures to give a sound foundation for the intuitive ideas discussed above.

The compatibility of DBMS and OS architectures is strongly influenced by allocation of the overall security policy to individual TCB subsets (i.e., the DBMS and OS subsets). In reality the DBMS subset is itself likely to be factored into more than one TCB subset.† The allocation of policy to the individual TCB subsets is then a critical factor in determining the overall efficacy of evaluation by parts. The conservative conclusions of the TDI are important and significant because they do not depend upon specific assumptions in this regard. But then they are also overly conservative with respect to particular policy allocations.

4 OS REQUIREMENTS

The TDI is of course an interpretation of the Orange Book. It therefore takes the Orange Book as a given. The Orange Book on the other hand was not written with the concept of evaluation of parts in mind. Therefore the Orange Book does not evaluate the features of an OS which make it easy or hard to build trusted subject DBMS's on top. As our examples above show the privilege features of an OS are extremely relevant. We therefore need additional criteria so as to be able to make the following statement: It will be easier to incrementally evaluate trusted subject

†This is true of practically all DBMS architectures proposed in the literature.

DBMS's built on top of OS A in comparison to those built on top of OS B. This is a difficult problem but one that must be confronted if products are actually going to be built and used.

5 SUMMARY

We have outlined some ideas for the critical and systematic analysis of evaluation requirements imposed by the coupling of various trusted DBMS and trusted OS architectures. The objective is to go beyond the conservative conclusions of the TDI by departing from its generic abstract perspective to consideration of

- concrete and realistic DBMS and OS architectures,

as well as

- specific assumptions regarding the allocation of the overall security policy to individual DBMS and OS TCB subsets.

We have also argued that the security community needs criteria to determine which OS architectures are more amenable to incremental evaluation of trusted-subject applications.

References

[1] National Computer Security Center. *Trusted Database Interpretation of the Trusted Computer Systems Evaluation Criteria.* NCSC-TG-021 (April 1991).

[2] Department of Defense. *Department of Defense Trusted Computer Systems Evaluation Criteria.* DOD 5200.28-STD (December 1985).