VERIFICATION TECHNOLOGY AND THE A1 CRITERIA

Position Paper by
Terry C. Vickers Benzel

The MITRE Corporation

Verification technology plays an important role in the development of secure computer systems that can satisfy the A1 requirements of the "Trusted Computer System Evaluation Criteria". I believe that the use of design verification techniques can be significant in developing secure systems. However, a number of issues need to be addressed before we can realize the greatest benefits from the use of the technology.

a. What are the A1 requirements for design specification and verification, and what are the necessary steps to take in order to meet these requirements?

   Despite the fact that the community has been applying the criteria to potential A1 systems (as I have, to the SCOMP system), I think this is very much an unanswered question. There are a number of requirements that are not well understood.

b. Can current verification systems be used to meet the A1 requirements for design specification and verification?

   If the answer to this question is no then we must try to determine where the weaknesses are and what needs to be done. It appears from examining existing verifications that there still is quite a bit of hand verification and hand waving going on. Does this imply that in practice A1 verification means hand verification ? If so what kind of assurance does that provide ?

c. If current verification systems can be used to meet the A1 requirements, then how should they be used ?

   Currently, each verification system has its own approach to design verification. There is a need for better definition and for consistency in approaches. This is not to say that all systems should employ the same approach. Certainly, each verification system should offer different benefits for differing applications.

d. How should future development of verification systems proceed so that they can be used to meet the A1 criteria ? Or, should the A1 criteria influence future development of verification systems ?

   Verification systems could be improved to help secure system developers meet the A1 requirements in the areas of formal models, covert channel analysis, and specification-to-code correspondence. I believe that the A1 criteria should influence the future development of verification systems. Perhaps this is unfortunate from the perspective of verification system developers. However, there is a real need for usable systems to perform security design verification.

The answers to these questions are not yet resolved; more research is needed into how existing verification technology can be used to meet the A1 criteria. There is an increasing government demand for systems that meet the A1 criteria. This means that there are developers trying to build A1 systems TODAY who can't wait for the next generation verification systems. Furthermore, design verification has been proven useful; I have seen a number of examples where formal specification and design verification has resulted in the discovery of security flaws, covert channels, and implementation errors, as described below.

THE HONEYWELL SCOMP DESIGN VERIFICATION

As team leader responsible for evaluating the formal design specification and verification of the Honeywell SCOMP system, I have worked closely with Honeywell in completing the formal verification evidence. In addition, as part of the evaluation, the team developed methods for mapping SPECIAL specifications to Pascal code and for reviewing Gypsy specifications.

The SCOMP system consists of a commercial Honeywell Level 6 minicomputer enhanced by a hardware Security Protection Module (SPM), security kernel software (approximately 10,000 lines of Pascal), and trusted software (approximately 10,000 lines of C). The security kernel design was formally specified and verified using the SRI International Hierarchical Development Methodology (HDM) and its tools. The design of the trusted software was formally specified and verified using the Gypsy Verification Environment. The approaches used and the results obtained from these two design verifications are discussed below.

The security kernel design was formally specified using SPECIAL. Then the MLS tool and Boyer-Moore theorem prover were used to verify that the top-level specification satisfied the information flow model embedded in the MLS tool. Informally, the SCOMP TCB was also shown to satisfy the access control model of Bell and LaPadula. In accordance with the A1 criteria, formal covert channel analysis and specification-to-code correspondences were also performed.

The trusted software design verification employed the Gypsy Verification Environment. The approach taken for this verification was to first provide informal (English-language) security requirements and a list of privileges, if any, for each trusted software function. The security requirements were derived from the security relevant portions of the B-level functional specifications. Then these security requirements and the security relevant behavior of each function was formally stated as a Gypsy top-level specification. Finally, lemmas were stated which corresponded to each security requirement. The verification then consisted of proving these lemmas using the Gypsy theorem prover. As was the case with the kernel verification, formal covert channel analysis and specification-to-code correspondences were also performed.

There were a number of positive results from these verification efforts. The specification effort resulted in the early discovery of security design flaws. The covert channel analysis resulted in the detection of covert channels that may not have been found through informal analysis. The specification-to-code correspondence played an important role in assuring that the implemented system corresponded to the verified design. Specification-to-code correspondence also contributed to the analysis of verification results and to the covert channel analysis. It is believed, by both the verifiers and the evaluators, that the formal design specification and verification effort led to a better implementation and increased assurance of the system's security.

In addition to the results mentioned above some not so positive observations can be made about the use of the verification technology. The amount of effort required to verify the SCOMP system was very large. The tools were often very slow, difficult to use, and unable to completely process a complex specification. There were many areas where tedious hand analysis had to be used. In particular, manual formula generation was required for the large modules, the analysis of the failed formulas produced by the Boyer-Moore theorem prover had to be done manually, and specification-to-code correspondence was a time consuming and tedious task for which little or no automated support exists.

CONCLUSION

Verification technology has been shown to be an important factor in the design of secure computer systems. However, as I have pointed out in this paper there are many areas where more research is needed. It is important that as the technology is used we address the issues I have raised, so that we can realize the greatest benefits from the use of this technology.