INTRODUCTION

The primary goal of the National Computer Security Center (NCSC) is to encourage the widespread availability of trusted systems. This goal is realized, in large measure, through the NCSC's Commercial Product Evaluation Program. This program focuses on the technical evaluation of the protection capabilities of commercially produced and supported systems. The standards against which products are evaluated are the Department of Defense Trusted Computer System Evaluation Criteria (TCSEC) for automatic data processing systems and the Trusted Network Interpretation of the Trusted Computer System Evaluation Criteria (TNI) for network systems.

The TCSEC and TNI classify systems into seven hierarchical classes based on features and assurances to support three types of security requirements - policy, accountability, and assurance. One of the assurance requirements, Design Specification and Verification, appears in the upper classes of the TCSEC and TNI. The highest level of trust, Verified Design or A1, requires that a Formal Top Level Specification of the design be maintained and shown, either formally or informally, to be consistent with the formal security policy model for the system. In addition, the requirements state that "[t]his verification evidence shall be consistent with that provided within the state-of-the-art of the particular Computer Security Center-endorsed formal specification and verification system used."

PURPOSE

The purpose of the Endorsed Tools List (ETL) is to inform system developers which formal specification and verification tools are endorsed by the NCSC for use in designing candidate A1 systems (that is, approved for use in satisfying the A1 Design Specification and Verification requirement). The ETL specifies the current version of each verification tool that is approved by the NCSC.

Additions to and deletions from the ETL will occur as the need arises, as determined by the NCSC. A compelling reason must exist to justify the addition of a verification tool onto the ETL. The proposed tool will have to offer some significant capability not provided by the current set of tools. Addition of a tool onto the ETL means that this tool may be named in a Verification Tool Memorandum of Agreement (MOA) between the NCSC and a system developer. The MOA should specify the specific tool and version to be used in the formal verification process. Likewise, profound changes in circumstances could result in the removal of a verification tool from the list (e.g., discontinuance of support from the tool developer). Removal of a tool from the ETL means that no *new* MOAs between the NCSC and system developers will be executed specifying that particular tool as the vehicle for formal specification and verification. Formal verification is a young technology. The tools are almost constantly undergoing enhancement and correction. Consequently, the version of a tool that has been evaluated by the NCSC may not be the newest most capable version. These modified versions of an endorsed tool are known as "beta" tool versions. Beta tool versions may undergo evaluation by the NCSC, and if the evaluation criteria are met, are declared to be the new evaluated version, are placed on the ETL, and replace the previously endorsed version.

Trusted system developers may choose to use a beta version of an endorsed tool. However, it must be realized that beta versions have not been completely evaluated and are not endorsed. Beta versions are released primarily to allow the verification tool developers to test the tool before submitting it for evaluation. Thus, beta tool users incur a slight risk. When entering a formal evaluation with the NCSC, specifications and proof evidence must be submitted which can be completely checked without significant modification using either the currently endorsed version of a verification tool or a previously endorsed version that was agreed upon in a MOA. Submitted specifications and proof evidence which are not compatible with the endorsed or agreed upon version of the tool may require substantial modification by the system developer.

The products on the ETL have been evaluated against the Formal Verification Environments Criteria (currently in draft). The outcome of the evaluation, including a complete analysis of the verification tool, is documented in a Technical Assessment Report (TAR). The ETL entry is intended as

only a brief summary for quick reference and therefore should not be used as the sole source for choosing a verification tool. Combining information from both the ETL entry and the TAR is vital to making a well-informed decision. The TARs can be obtained by contacting the Office of Research and Development at the National Computer Security Center in writing.

National Computer Security Center 9800 Savage Road Fort Meade, Maryland 20755-6000

ATTN: C33, TAR request

DEFINITIONS

BETA TOOL VERSIONS - In order to ensure the availability of the most current tool versions, tool enhancements to ETL tools may be released for production use during the "Beta test" stage of development. Beta tool versions may undergo evaluation by the NCSC, and if the evaluation criteria are met, are declared to be the new evaluated version, and are placed on the ETL. Vendors are free to use such Beta Tool Versions for production use.

ENDORSED TOOLS LIST (ETL) - A list composed of those verification tools that are currently recommended by the NCSC for use in satisfying the A1 Design Specification and Verification requirement in the Trusted Computer System Evaluation Criteria and Trusted Network Interpretation of the Trusted Computer System Evaluation Criteria.

FORMAL VERIFICATION ENVIRONMENTS CRITERIA - A document (currently in draft) that provides the basis for determining the technical merit and stability of formal specification and verification tools. The criteria provides evaluators with a metric with which to assess the breadth of a verification tool and provides guidance to manufacturers on the fundamental requirements of candidate endorsed verification tools.

TRUSTED COMPUTER SYSTEM EVALUATION CRITERIA (TCSEC) - A Department of Defense standard (DoD 5200.28-STD) that was published in December 1985 to provide a means for evaluating specific security features and assurance requirements available in "trusted commercially available automatic data processing systems". The rating scale in the TCSEC extends from one that represents a minimal level of trust to one for "state of the art" features and assurances.

TRUSTED NETWORK INTERPRETATION OF THE TCSEC (TNI) - A document that was published in July 1987 to provide a means for evaluating specific security features and assurance requirements as well as additional security services of networks. VERIFICATION TOOL MEMORANDUM OF AGREEMENT (MOA) - A Memorandum of Agreement between a trusted system developer and the NCSC in which both agree on the endorsed verification tool and version to be employed for the secure system development effort.

FORMAL DEVELOPMENT METHODOLOGY

VENDOR: Unisys Corporation

VERSION EVALUATED: 12.4

EVALUATION DATE: 15 January 1988

PRODUCT DESCRIPTION:

The Formal Development Methodology consists of a set of tools and languages, as follows:

(1) the Ina Jo language, for writing specifications and requirements

(2) the Inamod language, an extension of the Ina Jo language, for writing assertions about programs

(3) the Ina Jo Processor, for examining specifications and generating logical assertions; the latter state that the specifications meet the requirements

(4) Ina Flo, a multilevel security flow tool, and the Shared Resource Matrix (SRM) tools, both implemented as independent modules within the Ina Jo processor. The SRM generates a matrix for manual analysis and, unlike Ina Flo, requires no changes to existing Ina Jo specifications

(5) the Interactive Theorem Prover (ITP), for assisting users in proving the logical assertions generated by the Ina Jo Processor

(6) a Verification Condition Generator (VCG) for Modula for examining specifications and implementations and generating logical assertion; the latter state that the implementation meets the specification

(7) an ITP Post-Processor called Short, for generating transcript files of completed proofs. The post-processor eliminates all steps not actually used in a proof, converts the contents of the file from the ITP internal representation, and reformats the text

The goal of FDM is to design, formally specify and produce verified code for complete systems. However, some of the software tools of FDM are incomplete at this time. The Ina Jo and Inamod languages are considered complete; language extensions and new capabilities to aid the user are occasionally implemented. In addition, the Ina Jo language processor, the ITP, and the ITP post-processor are complete. The initial VCG, for Modula, and Ina Flo have not been completed and therefore were not part of the evaluation or endorsement.

FDM has been used extensively on over a half-dozen significant systems for DCA, AFWL, RADC, and others. Applications include Autodin II, the Secure Transaction Processing Experiment (STPE), a Job Stream Separator (JSS), a kernelized IBM VM (KVM), a Computer Operating System/Network Front End (COS/NFE), and the Secure Release Terminal.

EVALUATION SUMMARY:

FDM 12.4 has been evaluated by the National Computer Security Center. The evaluation team analyzed the changes between the 12.3 and Beta 4 version of the tool. The changes fell into the following categories: syntax changes, bug fixes, new features, and changes to theorems. In addition to the changes that were identified by the developer, the team examined configuration management evidence, source code, documentation, and existing problems with the tool that were identified in the Verification Assessment and Mitre Reports.

The evaluation was not exhaustive, but was sufficient to determine that version 12.4 is a distinct improvement over version 12.3. A few of the most notable enhancements are the increase in theorem proving capability, the addition of a type checking facility, and improved error reporting. A complete discussion of the changes to the tool can be found in the Technical Assessment Report.

GYPSY VERIFICATION ENVIRONMENT

VENDOR: Computational Logic Inc. (CLINC)

VERSION EVALUATED: GVE version 13.16 (Gypsy dialect 2.05)

EVALUATION DATE: July 1987

PROJECT DESCRIPTION:

The GVE consists of a set of tools and languages, as follows:

(1) the Gypsy language, which is used for both system design and specification

(2) the Verification Condition Generator, which translates specifications into problems in first-order logic

(3) the Gypsy Theorem Prover (GTP), for determining whether first-order logic statements are theorems

(4) the Gypsy to Bliss translator, which converts Gypsy code to Bliss code

(5) the Gypsy to Ada translator, which converts a subset of the present Gypsy language into non-standard Ada

(6) the DataBase Manager, which is used to track the status of various scopes and theorems the user is maintaining

(7) the Gypsy language parser, for determining the syntactic and semantic validity of a design

(8) an interface to the ZMACS editor for ease of text entry

(9) a program optimizer which proves the validity of efficiency-based decisions in compilation.

The Gypsy Verification Environment (GVE) is designed to be a complete verification system supporting both a specification language and a programming language. The goal is to formally specify and verify design and code for computer systems and applications. However, some of the software tools of the GVE are incomplete. The Gypsy to Ada translator currently converts a subset of the Gypsy language to nonstandard Ada. The dependency tracking mechanism of the GVE currently does not provide complete and proper tracking of dependencies between specifications, code, proofs and lemmas. The GVE has been used in a number of prior efforts, including the only A1 system (SCOMP), the largest examples of code verification (Encrypted Packet Interface), and is presently being used in several very large projects which are candidates for A1 evaluation.

The GVE has also received acceptance throughout the verification community, not only for its applicability and convenient approach, but also for its unique capabilities. The GVE is currently the only evaluated system which can handle concurrency or do code level proofs.

EVALUATION SUMMARY:

The Gypsy Verification Environment version 13.16 using Gypsy dialect 2.05 has been evaluated by the National Computer Security Center (NCSC) against the requirements specified by the Verification System Evaluation Factors. The NCSC has determined that the GVE satisfies all of the requirements at the minimum level. However, it was noted that there is an obvious need for better documentation for system users, especially for new users. In addition, the lack of a formal basis and formal semantic definition is a serious shortcoming that needs to be addressed in the near future. While the GVE is not without weaknesses, it has received acceptance throughout the verification community, not only for its applicability and convenient approach, but also for its unique capabilities. The GVE is currently the only complete system which can handle concurrency or do code level proofs. These are important features which make the system worth examining.

Verification Evaluation Factors

1. Specification Language 2. Theorem Prover 3. Implementation 4. Documentation 5. Features 6. Assurance and Soundness 7. User Interface 8. Methodology

For a complete description of how the GVE satisfies each of the evaluation factors, see Technical Assessment Report for GVE 13.16 using Gypsy 2.05.