TCX Project: High Assurance for Secure Embedded Systems

Thuy D. Nguyen, Timothy E. Levin, Cynthia E. Irvine
Naval Postgraduate School, Monterey, CA
{tdnguyen,levin,irvine}@nps.edu

Abstract

An overview of the Trusted Computing Exemplar (TCX) research project and its accomplishments to date are presented. The TCX project is constructing a separation kernel that will be high assurance and suitable for use in simple embedded systems. To guide the kernel development, we have created a reusable high assurance development framework. The main emphasis of this multifaceted research and development initiative is to transfer knowledge and techniques for high assurance trusted system development new developers, evaluators and educators.

1. Introduction[1]

Development of high assurance security products requires knowledge and techniques not commonly taught to or practiced by most software developers in the commercial sector. The lack of rigor and discipline in the software development process, driven by the focus on short time-to-market, performance and functionality, has produced rampant security vulnerabilities that gravely affect a large range of computing environments, from small deeply embedded safety applications to large enterprise software platforms.

This situation provides the impetus for the Trusted Computing Exemplar (TCX) project, whose goal is to provide an openly distributed worked example of how high assurance trusted computing components can be built [1]. Our approach to meeting this goal is described in Section 2.

It is our expectation that the open availability of the TCX results will enhance the capability to develop highly secure software in both commercial and open-source sectors. In particular, the high assurance development framework can be reused or adapted to support the development of secure systems that are more complex than the TCX demonstration system. The relevance of the TCX project is exemplified further by the surge of recent interest in high assurance systems, separation kernels, and evaluation profiles. [2, 3, 4]

This work-in-progress paper presents a brief overview of the TCX project and summarizes its current development status.

2. Project Description

The TCX project encompasses four related activities, an overview of which is presented in this section.

- Creation of a reusable high assurance development framework;
- Development of a reference-implementation trusted network component;
- Support for evaluation of the reference component against the highest assurance criteria as defined in the Common Criteria (EAL7) [5];
- Open dissemination of the results of the first three activities.

2.1 Reusable Development Framework

The TCX development framework consists of two major components: 1) high assurance life cycle framework and 2) high assurance rapid development environment (HARDE). The TCX life cycle model has augmented the spiral life cycle model [6] with the high assurance properties required by the EAL7 life cycle requirements. Rigorous configuration management and strict developmental security safeguards are part of the high assurance life cycle framework.

The TCX development environment consists of a documentation integration environment with which to construct and manage TCX project documents; development tools and procedures for construction of TCX software; and verification tools and procedures with which to determine with high assurance whether the system that is built is as was defined.

[1] This work was sponsored in part by the Office of Naval Research and the National Reconnaissance Office. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Office of Naval Research or the National Reconnaissance Office.

2.2 Reference Implementation

In general, kernel software is highly specialized and not well understood in the software development community. Adding high assurance security requirements to this mix increases the level of development effort exponentially. We have chosen to develop a high assurance separation kernel, along with a trusted demonstration application built to be hierarchically layered [7] on the Kernel, as a reference implementation for trusted computing.

The high-level requirements of the TCX Separation Kernel (TCX-SK) are: simplicity, compactness, portability, and an a priori assurance against system subversion. The primary security function of the TCX-SK will be to enforce process and data-domain separation, while providing basic operating system services sufficient to support specialized applications. The kernel will have a static runtime resource configuration and its security policy regarding access to resources will be based on static process/resource access bindings, which are subject to offline configuration. The kernel will implement static scheduling and support a small number of processes, data objects, and I/O devices. The kernel initialization component will bring the system into a secure initial state. All subsequent operations will be formally demonstrated to preserve the secure state.

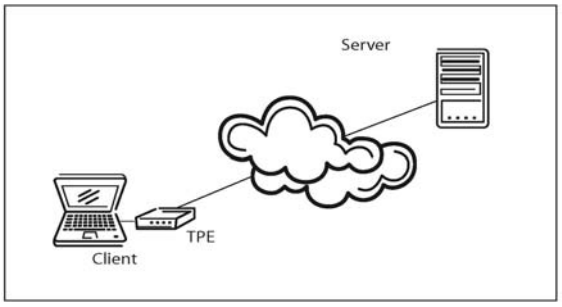

Figure1. MYSEA System Architecture: Client-TPE-Server

An application running on the TCX-SK will serve to demonstrate the kernel's utility and effectiveness. The application is the trusted path extension (TPE), which communicates security critical information between a user and a remote secure server, such as in the MYSEA distributed security architecture (see Figure 1 [8]).

2.3 Reference Component Evaluation

An independent security evaluation is required to provide confidence in the assurance claims made for a trusted component. Although the evaluation target for the TCX-SK is EAL7, currently there is not available a validated EAL7 protection profile for separation kernels. We have been working with the NSA as part of a team to develop an EAL6+ protection profile for separation kernels, which is in the draft form [4]. Our participation in the authoring of the separation kernel protection profile (SKPP) will aid the development of the Security Target specification [5] for the TCX-SK. Although the TPE application software will be designed and built to meet EAL6 requirements, there are currently no plans for its evaluation.

2.4 Open Dissemination

The outputs of this project will be openly available. This will include source code, specifications, and evaluation evidence and reports. By making available the various high assurance internal engineering specifications, evaluation and development framework documents, this project will provide previously unavailable how-to examples for high assurance trusted computing.

3. Project Status

The TCX project was initially conceptualized in late 2002 and project work was started in mid 2003. The feasibility of the effort was studied and it was determined that the static nature of the system, the simplicity of the policy to be enforced, as well as the considerable previous experience in high assurance development of our team made the project feasible within the timeframe envisioned. Since then, significant progress has been made in the following areas: threats and requirements analysis, formal model, life cycle management and development environment.

3.1 Threats and Requirements Analysis

The overarching concern addressed in very high assurance systems is the threat of subversion of the system itself by an adversary [9]. We have examined various threat models in the context of protecting high value information with a high assurance separation kernel, and derived from these threat models a set of security objectives and requirements, which were included in the SKPP. The SKPP only defines the minimal requirements for a class of high assurance separation kernels. Thus, we have completed the development of the top-level policy and requirements definitions specific to the TCX-SK. For Common Criteria evaluation, a security target

document is required that provides further detail about the implementation. We have produced a first draft of the TCX-SK Security Target.

3.2 Formal Model

The *separation kernel* concept was introduced in the early 1980s [10] and has since been used in a number of operational systems and research projects [11, 12]. In a separation kernel, the system entities are partitioned into *blocks*, which are to be kept separate from each other, with the exception of certain carefully controlled interaction channels. Despite the resurgence of interest in the separation kernel approach, the principle of least privilege [13] is often overlooked in the design of traditional separation kernels due to the belief that a separation kernel should only be concerned with partition-level resource isolation. Consequences of this omission include problems relating to all-or-nothing security and over-privileged programs.

We believe that to provide credible high assurance of policy enforcement, a system based on the separation kernel abstraction must be enhanced to support the principle of least privilege. A *least privilege separation kernel* can provide, in addition to the functionality and protection of the traditional separation kernel, a high level of confidence that the effects of activities caused by subjects (the system's active entities) may be minimized to their intended scope. This will make the system inherently more secure.

We have developed a formal presentation and discussion of the least privilege separation model that supports two orthogonal flow policies: 1) block-to-block flow control and 2) least privilege flow control between subjects and resources [14]. As the basis for a formal demonstration that the TCX-SK system enforces its security policy, we have implemented a preliminary version of this model with the Formal Development Methodology (FDM) tool set. In addition to an interactive theorem prover, FDM includes linguistic support for reasoning about the relationship between different levels of abstraction of a given system security property, for both state-based and transition-based properties.

3.3 Life Cycle Management

The TCX life cycle management process and configuration management (CM) scheme will satisfy the life cycle and CM assurance requirements defined in the SKPP. To date we have produced the following documents: Life Cycle Management Plan, Configuration Management Plan and Procedures, Personnel and Physical Security Plans. The Life Cycle Management Plan is the overarching document that defines policy, process and procedures to guide the TCX development and to ensure that the development is compliant with EAL7 assurance requirements.

The objectives of the CM plan include: ensuring the integrity of the configuration items, tracking changes to the configuration items, and ensuring that only authorized changes are made to the configurations items. Items to be managed under this plan include all documents, source code, specifications, and other items written, used or developed (including bug reports, security flaws, and CM and development tools) as part of the product development process.

Because of the nature of the project, integrity is the primary policy concern of the Personnel and Physical Security Plans, though confidentiality is not disregarded (e.g., with respect to unintended dissemination of incomplete project items). These security plans define personnel policies and physical protection policies necessary to ensure the confidentiality, integrity and physical protection of the TCX during its entire life cycle.

Another accomplishment is the establishment of a CM system that is physically isolated from the TCX development network and our campus network. The CM system is administered and operated by a CM team whose members are not part of the TCX development team. The Perforce software configuration management tool was selected for this project based on the analysis of available CM software [15].

3.4 Development Environment

As part of our research, we are developing an automated documentation referencing system that can be used to facilitate traceability and correspondence between requirements specifications (both evaluation and engineering) and the TCX-SK implementation (both software and hardware). We first developed the Documentation Development Standards document that describes the policy and process for authoring TCX documents. Then we established the high-level requirements for an XML-centric Documentation Integration Environment (DIE). The specifications and implementation of the initial version of the DIE have been completed, and the DIE is in use by the project team for the creation and requirements mapping of project documents.

We have also completed both the Software Development Standards document and the design and construction of the prototype TCX development network. The Software Development Standards document includes (1) processes for designing, approving and developing Configuration Items in conformance with the CM Plan, and (2) the TCX coding standards. The TCX development network is built upon the NPS "Intranet" infrastructure. It utilizes Virtual Private Network (VPN) technology to provide developers with the security and flexibility to work from different locations within the campus.

To meet EAL7 requirements, a TCX verification environment was developed in parallel with the software development environment. The TCX verification tool requirements were defined, and the verification server and several candidate tool sets were assembled, including FDM, PVS, and ACL2. Investigation and experimentation is ongoing to determine suitability of each tool suite for the verification of high assurance systems models and specifications.

4. Conclusion

This paper describes the TCX project and its accomplishments to date. During the first phase of research, we have established a working and reusable life cycle and development framework. The next research phase will include the architectural design and initial implementation of the TCX-SK and TPE application, the development of a web-based dissemination system, and construction of the formal specifications required for an EAL7 evaluation.

References

[1] Irvine, C. E., Levin, T. E., Nguyen, T. D., and Dinolt, G. W., "The Trusted Computing Exemplar Project," *Proceedings of the 2004 IEEE Systems, Man and Cybernetics Information Assurance Workshop*, West Point, NY, June 2004, pp. 109-115.

[2] "Safe Critical Products: INTEGRITY-178B RTOS," Green Hills Software, Inc., http://www.ghs.com/products/safety_critical/integrity-do-178b.html.

[3] "OS Security: Secure Operating System and Architecture," LinuxWorks, Inc., http://www.lynuxworks.com/solutions/security.php.

[4] "U.S. Government Protection Profile for Separation Kernels in Environments Requiring High Robustness," National Security Agency, 1 July 2004, http://niap.nist.gov/pp/draft_pps/pp_draft_skpp_hr_v0.621.html.

[5] *Common Criteria for Information Technology Security Evaluation*, Version 2.2, CCIMB-2004-01-00[1, 2, 3], January 2004.

[6] Boehm, B. W., "A Spiral Model of Software Development and Enhancement," *IEEE Computer*, Vol. 21, Issue 5, pp. 61-72, 1988.

[7] Dijkstra, E. W., "The Structure of the "THE"-Multiprograming System," *Communications of the A.C.M.*, Vol. 11 No. 5, pp. 341-346, 1968.

[8] Irvine, C. E., Levin, T. E., Nguyen, T. D. Nguyen, Shifflett, D. Khosalim, J., Clark, P. C. Wong, A. Afinidad, F., Bibighaus, D., and Sears, J.,"Overview of a High Assurance Architecture for Distributed Multilevel Security," *Proceedings of the 2004 IEEE Systems, Man and Cybernetics Information Assurance Workshop*, West Point, NY, June 2004.

[9] Anderson, Emory A., Irvine, Cynthia E., and Schell, Roger R., "Subversion as a Threat in Information Warfare," Journal of Information Warfare, Volume 3, No.2, June 2004. pp 52-65.

[10] Rushby, J., "Design And Verification Of Secure Systems," *ACM Operating Systems Review*, 15(5), 1981.

[11] Adams, C., "Real-Time Operating Systems and Hardware Support," *Avionics Magazine*, May 2003. www.aviationtoday.com/reports/avionics/previous/0503/0503real_time.htm.

[12] Chincheck, S. J., "Programmable Embeddable INFOSEC Product," http://www.nrl.navy.mil/content.php?P=03REVIEW159.

[13] Saltzer, J. H., and Schroeder, M. D., "The Protection of Information in Operating Systems," *Proceedings of the IEEE*, 63(9):1278-1308, 1975.

[14] Levin, T. E., Irvine, C. E., and Nguyen, T. D., "A Least Privilege Model for Static Separation Kernels," Technical Report NPS-CS-05-003, Center of Information Systems Security Studies and Research, Naval Postgraduate School, October 2004.

[15] Zeigenhagen, L., "Evaluating Configuration Management Tools for High Assurance Software Development Projects," Masters Thesis, Naval Postgraduate School, June 2003.