A Case Study in Security Requirements Engineering for a High Assurance System

Cynthia E. Irvine, Timothy Levin, Jeffery D. Wilson[‡], David Shifflett, Barbara Pereira
Department of Computer Science
Naval Postgraduate School
Monterey, California 93943
[irvine, levin, shifflet, pereira]@cs.nps.navy.mil

[‡]United States Marine Corps
WilsonJD@mcsc.usmc.mil

ABSTRACT

Requirements specifications for high assurance secure systems are rare in the open literature. This paper presents a case study in the development of a requirements document for a multilevel secure system that must meet stringent assurance and evaluation requirements. The system is secure, yet combines popular commercial components with specialized high assurance ones. Functional and non-functional requirements pertinent to security are discussed. A multidimensional threat model is presented. The threat model accounts for the developmental and operational phases of system evolution and for each phase accounts for both physical and non-physical threats. We describe our team-based method for developing a requirements document and relate that process to techniques in requirements engineering. The system requirements document presents a calibration point for future security requirements engineering techniques intended to meet both functional and assurance goals.

1. INTRODUCTION

Sometimes an organization possesses information of such a critical nature that its inappropriate exposure or modification would cause grave damage to the enterprise. Because networked computer systems are now used to store and manage this sort of highly sensitive information, management may seek assurances that those computers and networks cannot be misused in ways that result in lapses of security policy enforcement that would expose the organization to unacceptable risk. Malicious software has long been understood to be a significant threat to information security [41]. Multilevel secure systems are intended to control the sharing of highly sensitive information in the face of malicious software in a manner commensurate with policies relating to information confidentiality and integrity [13].

The views expressed in this paper are those of the authors and should not be construed to reflect those of their employers or the Department of Defense.

High assurance computers and networks are intended for the protection of critical information resources in environments that involve access by individuals with a range of authorizations to enterprise information. A high assurance system is intended to be an implementation of the Reference Monitor Concept [7], a notion that describes the ideal security policy enforcement mechanism. This ideal has three characteristics. First, it is always invoked to perform its enforcement duties. Second, attackers cannot successfully penetrate the mechanism and thus cause it to fail. Finally, it is small enough to ensure that it will, in fact, correctly enforce policy and that it contains no artifices that might be used to neutralize the policy enforcement mechanism. One objective of high assurance security engineering is to address the threat of system subversion [26]. Examples of system subversion abound in commercial software as is evident from web sites devoted to *Easter Eggs* [5]. That subversion can be cleverly implemented so as to make it virtually impossible to detect was amply illustrated by Thompson [38].

A high assurance system is a real instance that approaches the Reference Monitor Concept ideal through a rigorous security engineering process. Its development may begin with the capture of the security policy to be enforced and an interpretation of that policy in terms of a computer system. This may produce a formal security policy model and subsequent evidence that policy enforcement objectives are met. In parallel with that formal approach, the engineering team will develop a series of specifications that will move from high level requirements to detailed implementation documents and code. The system requirements specification is the most abstract of this latter set of documents. A system requirements specification for a secure system will incorporate security consideration in conjunction with all other requirements. Such a requirements document provides a good example for those studying security requirements concerns.

In many cases, the design and development of high assurance secure systems has taken place behind closed doors [30, 3, 32]. Although these systems may have been evaluated under the guidelines of various criteria intended to assess their security functionality and lifecycle assurance, little documentation is available for public scrutiny beyond the final evaluation reports published at the end of the evaluation

process. The actual documents produced during system design and development remain cloaked in proprietary secrecy and government constraints and non-disclosure agreements.

Three of the authors of this paper have been members of engineering teams whose work resulted in the design and implementation of a commercially available highly trusted secure system [32]. Since entering a less restrictive environment, we and our collaborators have embarked on the development of a high assurance network that relies upon commercial-off-the-shelf (COTS) hardware and software to present the user interface. This paper is intended to describe the general method used to develop the high level requirements specifications for a high confidence secure system.

Using our MLS LAN System Requirements Document (see Appendix) for the high assurance multilevel secure local area network as a backdrop and example, we will outline the steps needed to move from an intuitive, *ad hoc* notion of the system requirements to a coherent document that can be used for more detailed system specifications and design. The security requirements specification captures not only the system's functional requirements but the non-functional requirements as well.

The remainder of this paper is organized to illustrate our requirements development method. To put the requirements specification process in the proper context, background information about the multilevel secure LAN will be provided in Section 2. Techniques used to create the specification will be discussed in Section 3 with reference to Appendix A, an image of the requirements specification document. This section will present our multi-dimensional threat model as well as the method we used to develop the specification. Some general observations about high assurance security requirements specifications and recommendations for process improvement follow in Section 4. Our presentation concludes in Section 4.2 with a summary.

2. OVERVIEW OF THE MULTILEVEL SECURE LAN

The Naval Postgraduate School (NPS) Multilevel Secure Local Area Network (MLS LAN) project [21, 16, 15, 8] is an effort to develop a high assurance LAN that leverages COTS components. The project was driven by two overarching requirements. The first was to protect security-critical information from unauthorized disclosure or modification. The second was to provide users with the ability to use popular COTS office productivity tools to organize and manage information at authorized sensitivity levels.

To achieve these objectives, we undertook the design and development of a prototype infrastructure for applications: a local area network that incorporated both high assurance security policy enforcement mechanisms and COTS software and hardware platforms for users' desktops. The former was intended to leverage the considerable government investment in highly trusted systems, while the latter objective resulted from the realization that unless a secure system offered users the same sort of convenient interfaces users had come to expect when handling *normal* information, the secure system would fail due to lack of *user acceptability* [36].

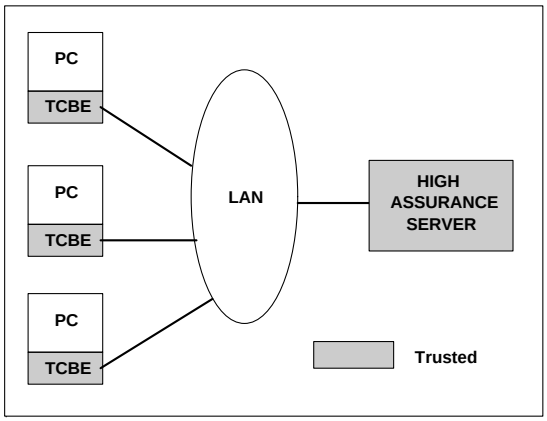

Figure 1: NPS Multilevel Secure LAN Architecture

The NPS MLS LAN project is intended to be a system that can provide controlled sharing of labeled information while permitting users to access that information through popular PC-based COTS personal and office productivity applications. Its architecture is illustrated in Figure 1 [15].

The high assurance server enforces the security policy and controls access to information. It is the core trusted computing base (TCB) for the distributed system. Application protocols run on the High assurance server and provide services and access to shared resources. Each PC is to be equipped with a trusted computing base extension (TCBE) plug-in board that will provide TCB support at the workstation. From these clients, users log on to the TCB, establishing an identity for audit and access control purposes. The TCB components, either the high assurance server base or the TCB extensions, were the only components directly connected to the physical network. Individual components are discussed below.

2.1 Trusted Servers

Each Trusted Server consists of the high assurance TCB, which enforces critical security policy, and untrusted application server instances (viz. one per security level per user) constrained by the TCB. The server supports sharing and labeling and is functionally equivalent in terms of overall application-level protocol support to a COTS application server for the particular protocol it is providing. Thus, it is compatible with existing COTS client packages.

Among the application servers we have adapted to the high assurance environment are: Internet Mail Access Protocol (IMAP) [2] based on a port of the University of Washington IMAP server [17], Hypertext Transfer Protocol (HTTP) providing an Apache-based port [11], and Simple Mail Transfer Protocol (SMTP) based upon sendmail [14]. The servers required little or no code modification to be adapted to the multilevel environment. With a proper configuration, users can view information at or below their current session levels.

2.2 High Assurance Base

The Wang XTS-300 provides our high assurance base [31]. This TCB, by virtue of the protection domains it creates,

provides confidence that malicious code will neither cause the exfiltration of sensitive data nor the corruption of information of higher integrity. Thus, one has confidence of correct security policy enforcement.

The high assurance server is defined by the broad properties needed for a viable commercial product. Our definition of a high assurance base is a TCB on the Evaluated Products List (EPL) [1] with a Class B3 or higher digraph based upon an evaluation against the Trusted Computing System Evaluation Criteria (TCSEC) [28] or its network interpretation [29], or equivalent Common Criteria requirements [4].

Modifications to XTS-300 TCB networking interfaces contribute to the support of the following desired functions: (1) a trusted path between client workstations and the XTS-300, (2) session-level negotiation at the XTS-300 from the client workstations, and (3) single-level session communications on the Ethernet for client workstations at different session levels. Our modifications permit multiple clients at different access classes to communicate with the server through a single physical network device [15].

2.3 Client Workstations

Client workstations are typical COTS PCs hosting a popular commercial operating system and a commercial application suite. For mail services the clients include: Lotus Notes, Outlook, Pine, Postal, and Netscape [1]. A typical browser supports the client interface to web pages.

To insure that object reuse requirements would be met, workstations are considered to be, in effect, "diskless," with sufficient volatile RAM-disk capability to support a wide variety of user applications. The workstation TCB extension will satisfy object reuse requirements by ensuring that RAM and other volatile primary and secondary storage at the workstation are purged with each change of session level or new user login at the workstation.

2.4 Trusted Computing Base Extension

To extend the TCB across the network, the architecture includes a trusted computing base extension (TCBE) at each COTS workstation [19, 8, 39]. This component is planned to provide the following services:

- A secure attention key (SAK) that permits users to establish unambiguous communication with the high assurance TCB for unspoofable presentation and capture of security critical data at the user interface.
- Non-bypassable, controlled access to the LAN.
- Protected communication channels between the TCB and the TCBE. These protected communications are based upon protocols that support both the establishment and maintenance of a trusted path, and session-level communications.
- Mechanisms to ensure high assurance object reuse at the client PC for both primary and secondary storage. Experiments by Agacayak [6] indicate that this is possible from an add-on card.
- Control of the client and its resources at the time of boot and control over security critical actions throughout the client session.

[1]These application names: Lotus Notes, Outlook, and Netscape, are trademarked by their respective owners.

We concluded that a carefully written and reviewed system requirements document was needed to guide the development of the system architecture and various communications protocols needed in the trusted LAN.

3. REQUIREMENTS SPECIFICATION

In this section we will provide some insights into our system requirements specification process. The motivation for the process is described. Functional and non-functional requirements are discussed. Our multidimensional threat model is presented. Finally we describe our requirements specification method.

3.1 Motivation and Objectives

A principal motivation for producing a system requirements specification was to provide focus for a complex effort. As we began exploring the design of a group of distributed mechanisms that had to result in coherent enforcement of the network security policy, it became clear that unless we had the same conceptual system as a goal, the result might be ineffective. In order to create a functional specification from which we could proceed, a requirements document that could be shared among all members of the engineering team was needed.

Another driver for the process was the desire to have a tool that would permit us to be able to judge when we were done building the LAN. The students were particularly concerned about *requirements drift* and *requirements inflation* [12] that might have created a Sisyphean task. The requirements specification also allowed us to examine the system in the abstract to determine whether some essential element had been omitted.

Several of us viewed requirements specification development for high assurance secure systems as a social process within the context of a larger engineering design and development effort and wanted to teach this aspect of high assurance system requirements specification to a new generation. We are not aware of any books or tools that can convey the social and interactive nature of this process as well as a "laboratory exercise".

The objective of the requirements specification process was to produce a high level yet rigorous description of system behavior that is both complete, consistent, and well formed. Our project objective was to produce a framework for an enabling technology to both support desired user applications and enforce the security policy. The requirements specification describes what we are trying to do, not how we are going to do it.

3.2 Functional and Non-Functional Security Requirements

Some state very broadly that all security requirements are non-functional [27, 40]. It has been suggested that functions implement system state changes and that *functional requirements* define how those functions will behave and what states can be reached [33]. In this case, certain security requirements **are** functional, as they affect changes in state. Consider for example an attempt to gain access in a system enforcing a mandatory confidentiality policy. Models for implemented systems [9, 10] describe system state changes while preserving system security properties. While the state of the system is being functionally transformed, *non-functional requirements* ensure that a set of orthogonal system properties are maintained.

To illustrate the nature of functional and non-functional requirements, consider two systems: a system enforcing a mandatory confidentiality policy (designated SECURE) and a system to automate traffic lights (designated TRAFFIC). Table 1 illustrates a few requirements for each. For the traffic light system, the requirement that the traffic light turns yellow after it has been green for three minutes is a state changing requirement and is functional. The accuracy of the clock that must be used to measure when three minutes have elapsed presents a non-functional, non-state changing requirement. For the secure system, users must present a valid password in order to log in to the system. This is a requirement which must be met in order for a state change from *user-logged-on* to *user-logged-in* to occur. Similarly, the requirement that secret information be stored in objects labeled SECRET affects whether information can move from an unstored to a stored state. In contrast, the requirement for the accumulation of an audit trail recording all access attempts does not affect system state. When access attempts succeed, state changes and when they fail state is unchanged. In either case, an audit record is generated.

Table 1: Functional and Non-Functional Requirements

Type	Requirement	System
Functional	A user shall present a valid password in order to log in to the system.	SECURE
	Secret information shall be stored in objects labeled SECRET.	SECURE
	The traffic light shall turn yellow after it has been green for three minutes.	TRAFFIC
Non-Functional	Audit records shall be recorded for all access attempts.	SECURE
	The accuracy of the clock shall be to within five seconds.	TRAFFIC

Thus we conclude that the construction of a high assurance system intended to enforce national security policy imposes both functional and non-functional requirements on a system. However, our requirements specification is not organized to distinguish functional from non-functional security requirements.

3.3 Threat Model

The system requirements specification is motivated by various classes of threats. Starting with an understanding of the elements of the engineering process [12, 37], we needed to understand how our threat model could best be represented and addressed. Figure 2 illustrates our modification to the requirements process. An understanding of the threat model within the system's developmental and operational domains will drive system requirements. As stated in Section 2, the objective of the system was to enforce a policy regarding the disclosure and modification of sensitive information. Availability was not a primary concern, as is reflected in the threat model.

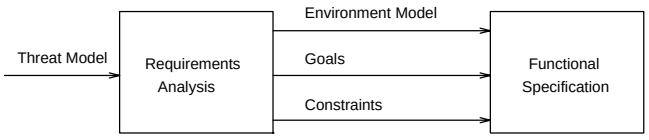

Figure 2: Requirements Process

The threats to be addressed are of two kinds, representing the domains of concern. Developmental threats form the first major class of threats to the TCB. These are threats to the ability of the system to fulfill its requirements. There are many design and implementation mistakes that can occur during a standard software development project [23]. The threat of subversion [26] is a major concern in the development of secure systems as the objective is to construct a system that is self protecting [36]. To counter mistakes due to poor engineering, we use a rigorous engineering process. To counter subversion, we use configuration management and quality assurance.

Development threats are illustrated in Figure 3. Interfaces to the TCB are shown and threats utilizing those interfaces are illustrated in hexagons. These threats include both physical and non-physical attacks. It is worth noting that adequate configuration management and quality assurance can mitigate many physical threats during the system development phase. Additional measures to address physical threats include traditional disaster recovery methods [34] such as: fire and earthquake protection, off-site backups, and locked offices and laboratories.

A comprehensive framework is needed during the development phase of a secure system to counter developmental threats. Requirements specification for this framework are beyond the scope of the example in Appendix A, but some guidance for high assurance development appears in standard evaluation criteria [28, 4].

Although our developmental threat model is principally concerned with eliminating subversion that would affect the ability of the system to enforce its confidentiality and integrity policies, a benefit of good engineering and of the configuration management and quality assurance process is the reduction of possible flaws that might result in denial of service from within the TCB itself.

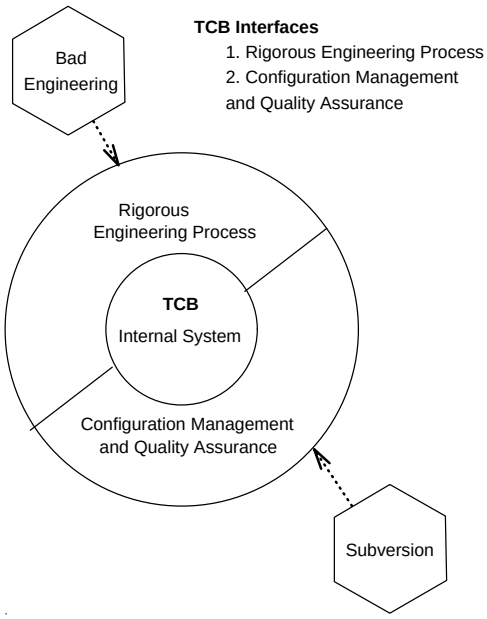

Figure 3: Development Threats

The second, and more obvious, class of threats is operational threats. These are the threats that the system is designed to counter and that come to mind when one thinks of system security. In the case of the MLS LAN, a high assurance multilevel system, the threats fall into three classes: network threats, malicious software, and misbehavior. Network threats are attacks to the communications protocols within the LAN. For example, an attempt could be made to use a non-TCBE equipped workstation attached to the LAN to modify or collect communications traffic. Malicious software is the principal operational threat to the server. This software would attempt to violate security policies for information confidentiality or integrity by obtaining unauthorized access to information. Trojan Horse software represents a classic example of malicious software and can be used to either directly or indirectly access information [22].Misbehavior applies to the MLS LAN Workstation where both user actions and malicious software may be used as attack paths. In this case, either users or software attempt to nullify the TCB Extension in order to gain unauthorized access to protected information.

We further decompose the operational threats into software threats and physical threats. Figure 4 illustrates the system of interest, i.e., the trusted computing base (TCB), its interfaces, and the operational software threats to the system by external elements that will drive the requirements specification. Thus the TCB is the internal system. Its interfaces are to the communications protocols, application proctol server, and MLS LAN Workstation. (A direct correspondence between Figure 4 and the second figure of the requirements document (MLS LAN Component Overview) can be observed.) External threats are network attacks, malicious software and misbehavior. The requirements document, when complete, addresses these threats. For example in the requirements document, the section on MLS LAN Connection Protocol Requirements addresses mitigation of the threat of network attacks on the system.

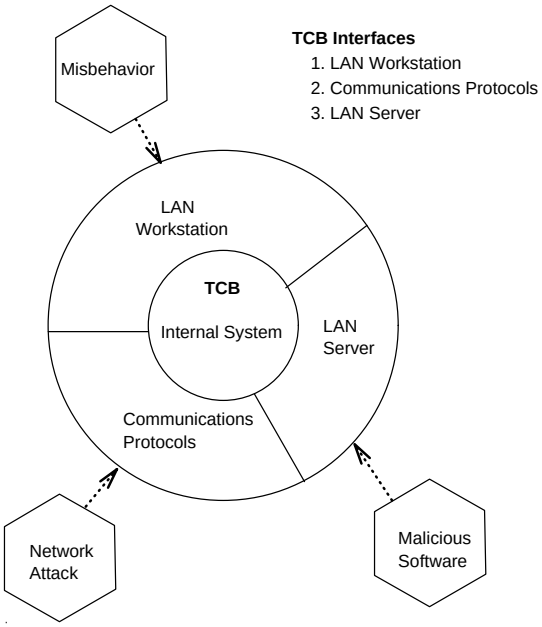

Figure 4: Operational Software Threats

An often neglected aspect of system lifecycle management that requires attention is that of physical access to and tampering with the system. For example, if an individual were to break the seals on the XTS-300 or the TCBE and replace the chip containing the BIOS, this would render the system insecure. Added to the usual configuration management concerns was that of physical protection of the fielded trusted system components. Thus, physical threats complement software threats to the operational system, and are illustrated in Figure 5. The TCB and its interfaces are the same, but the external threats are tampering with the communications lines, and TCB components, i.e., the XTS-300 or the TCB Extensions.

In summary, our threat model accounts for the developmental and operational stages of system evolution and for each stage accounts for both physical and non-physical threats. It is clear that denial of service is not addressed by the threat model for this project; however, the customer objective was not the enforcement of an availability policy. Nothing will prevent a user from monopolizing workstation resources; flaws in commercial workstation operating systems or applications or in server software could result in loss of availability; and physical attacks such as cutting communications links or destroying computers are guaranteed to deny service.

3.4 Method

Using a requirements specification format [20], we started with a skeletal requirements specification and filled in sections. Often we found that requirements from one section would affect decisions associated with another aspect of the system. Thus the requirements specification process was iterative within the component.

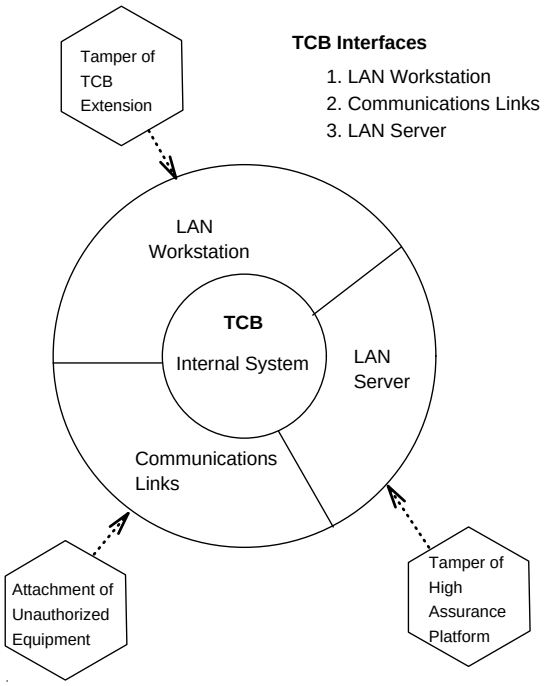

Figure 5: Operational Physical Threats

We used a two-level waterfall process [35] to ensure that the requirements specification was realistic. After completing what appeared to be a reasonable draft of the system requirements specification, we moved to the development of a specification for various communications protocols and protection mechanisms of the distributed TCB. (Drafts of both the requirements specification and the functional specification appeared as appendicies to a student thesis [42].) This was a much more detailed and concrete design statement that provided insights into the implementation details we were working toward. Those realities fed back into the system requirements document in interesting ways.

1. Design decisions gone awry sometimes indicated the need for a guiding principle at the requirements level.
2. Fundamental requirements statements moved out of design and up to requirements.
3. Conflict with real world possibilities resulted in clarification or refinement at the requirements level.

Concurrent development of a functional specification allowed us to identify notions that were generalizable and could be abstracted for inclusion in the requirements document. Conversely, items more appropriate for the functional specification were removed from the requirements document. For the less experienced members of the team, the temptation to include implementation details is enormous. It is fun to think about the implementation and harder to describe the abstract system. On the other hand, feedback from the more detailed design phase plays a key role. This feedback approach permitted us to develop documents that would be suitable for evolutionary engineering processes [24, 25] as experiential or environmental factors lead to requirements for new versions of the system.

Abstraction is one of the principal objectives of a high level requirements specification and one of the most difficult to achieve. Although our specification is intentionally abstract it is not intended to be vague, but instead semantically precise. There should be no ambiguity with respect to requirements that the system absolutely had to meet. In addition, the specification should be sufficiently abstract that a variety of implementations could satisfy the requirements [33]. For example, our specification states in Section 3.2.1.1 that "the TCB shall provide a Secure Attention Key (SAK) mechanism to invoke a trusted path from workstations to which the TCB has been extended." Nowhere is there a statement regarding how that SAK is to be implemented: that is the purview of the design team. It could be invoked from a keyboard using a combination of keys such as "CTRL-ALT-DEL" or it could be invoked using a special button.

Another aspect of the system that is abstracted away is the functionality associated with the applications. They are relevant to the framework under consideration only in so far as they impose requirements for support from the network in the form of resources for processing capability, memory and network bandwidth.

A key part of the process of developing the requirements specification involved recognition of implementation details. Through a process of winnowing, we were able to avoid inclusion of the implementation details. Some detailed descriptions were transferred to nascent design documents; some discarded; and others were generalized and abstracted to become true requirements. For example, instead of stating what application protocol services were to be supported, the specification states in Section 3.1.3 that "the MLS LAN shall provide...application protocol services ...".

A fundamental concept for requirements development is that of completeness. In reviewing progress and draft versions, tests for completeness can include *gedanken* exercises, such as: "Can a useless or insecure system be built to this abstract specification?" A mapping of the requirements specification to the threat models is used to ensure that all threats are accounted for. The individual and group reviews ensured that these questions were asked regularly throughout the requirements development process.

3.4.1 Social Process in Requirements Specification

Members of our specification team represented multiple stake holders. The most senior members of the team represented the customers, while junior members of the group represented those whose task would be system implementation. Ultimately, system implementation is the job of the entire team, however during the specification process, it is necessary to consider the perspectives of a larger set of individuals. The customer must be represented because if ignored, the system might become an amusing sandbox for implementation team experimentation, yet the likelihood that it would actually be used would be substantially reduced. The engineering team must be represented because this is the group that must be able to move from the abstract specification to a concrete implementation. Certifiers and accreditors

must be represented because they are the ones responsible for attesting that the implementation is faithful to the specification. Various sets of system users are represented and include typical users, operators, and administrators. Our experience was that the active participation of all stakeholders helped to ensure that the specification was well balanced with respect to the various and sometimes conflicting views of these stakeholders.

For several months we met on a weekly basis to discuss each new draft of the document. This type of teamwork is best achieved in an environment in which criticism is viewed not as a subjective attack on the writer but as an objective, scientific attempt to achieve the best possible result. As the team matured criticism was accepted and members with strengths in particular areas were able to contribute to the emerging document.

The team approach adds to the assurance of the resulting system and mitigates development threats. Because a group of system architects inspected the documents and discussed its semantics, the addition of a subversive artifact during the design stage is considerably more difficult [26]. This threat is directly addressed by way of the project's configuration management and quality assurance system.

3.4.2 Implicit Requirements

Inspection of the MLS LAN System Requirements Document shows that a number of the functional requirements are quite general. The system must support application protocols, but it is not necessary to specify which protocols. The system must support inexpensive commercial PCs as user workstations. It must support up-to-date versions of commercial operating systems. It follows implicitly that client applications can be up-to-date. The latter two requirements encompass not only a requirement for immediate system performance, but also one for adaptability in that both the COTS operating system and the applications it supports must be upgradable at any time.

4. SUMMARY AND CONCLUSIONS

When attempting to build a high assurance system, the objective is to provide a high level of confidence that security policy will be correctly and continuously enforced. Minimization has been recognized as an effective means of ensuring that the system can be judged for correctness and completeness [18]. In a minimized system we ask two questions. First, are all components within the system boundary, i.e., the Trusted Computing Base perimeter, necessary for the correct enforcement of security policy, and, second, are the mechanisms organized such that they are sufficient for security policy enforcement. If additional functionality is added to the secure system, then additional effort will be required to demonstrate that the result meets the assurance requirements. The task in developing a requirements document is to restrict the design of the system by providing sufficient detail so that system specifications are well focused.

4.1 A Calibration Point

Over the past decade there has been significant work in the area of requirements engineering [40] and new efforts are underway to extend these achievements to the area of security requirements engineering. What metric is to be applied to judge the effectiveness of these new security requirements engineering techniques? Will those requirements be sufficiently abstract to permit the development of a wide range of system implementations? Will those requirements introduce restrictions that will facilitate the development of system specifications and other documents [33]? The requirements document provided here is a worked example in that it has provided both the abstraction and restrictions necessary to shorten the effort required to specify and construct a high assurance multilevel system. Thus, our specification can provide a calibration point for new techniques in requirements engineering.

4.2 Conclusions

This paper has presented a case study in the development of a requirements document for a multilevel secure system that must meet stringent assurance requirements. The system is secure, yet combines popular commercial components with specialized high assurance ones. We described security objectives as having both functional and non-functional requirements. A multi-dimensional threat model that accounts for developmental and operational phases of system evolution and considers both physical and non-physical threats has been presented. Explicit consideration of physical threats to the system are addressed in the requirements engineering process. We have described our team-based approach to system specification and design. By assuming the views of various stakeholders in the system, through open, nonjudgemental discourse, and by using the threat model and the high level design specifications as a check, we have developed an abstract requirements specification. The specification of the NPS MLS LAN provides a worked example of a requirements document for a high assurance secure system and thus may be unique in the open literature. Our specification can be used as a calibration point for future security requirements engineering techniques intended to meet both functional and assurance goals.

5. ACKNOWLEDGEMENTS

The authors wish to thank Professor Luqi, of the Naval Postgraduate School Computer Science Department, for useful discussions of the requirements engineering process. The authors are grateful to their U.S. Navy and government sponsors for their support of this research.

6. REFERENCES

[1] Evaluated Products List, National Computer Security Center. http://www.radium.ncsc.mil/tpep/epl/.

[2] IMAP Information Center. http://www.washington.edu/imap/.

[3] Gemini Trusted Network Processor (GTNP). In *Information Systems Security Products and Service Catalog Supplement*, Report No.CSC-PB-92/001. April 1992. 4-SUP-3a.3.

[4] ISO/IEC 15408 - Common Criteria for Information Technology Security Evaluation. Technical Report CCIB-98-026, May 1998.

[5] *The Easter Egg Archive*. http://www.eeggs.com/, last modified 19 May 2000.

[6] C. Agacayak. TCBE Control of Object Reuse in Clients. Master's thesis, Naval Postgraduate School, Monterey, CA, March 2000.

[7] J. P. Anderson. Computer Security Technology Planning Study. Technical Report ESD-TR-73-51, Air Force Electronic Systems Division, Hanscom AFB, Bedford, MA, 1972. (Also available as Vol. I,DITCAD-758206. Vol. II, DITCAD-772806).

[8] S. Balmer. Framework for a High-Assurance Security Extension to Commercial Network Clients. Master's thesis, Naval Postgraduate School, Monterey, CA, September 1999.

[9] D. E. Bell and L. LaPadula. Secure Computer Systems: Mathematical Foundations and Model. Technical Report M74-244, MITRE Corp., Bedford, MA, 1973.

[10] D. E. Bell and L. LaPadula. Secure Computer System: Unified Exposition and Multics Interpretation. Technical Report ESD-TR-75-306, MITRE Corp., Hanscom AFB, MA, 1975.

[11] E. Bersack. Implementation of a HTTP (Web) Server on a High Assurance Multilevel Secure Platform. Master's thesis, Naval Postgraduate School, Monterey, CA, December 2000.

[12] V. Berzins and Luqi. *Software Engineering with Abstractions*. Addison Wesley, Reading, Massachusetts, 1990.

[13] D. L. Brinkley and R. R. Schell. Concepts and Terminology for Computer Security. In Abrams, Jajodia, and Podell, editors, *Information Security: An Integrated Collection of Essays*, pages 40–97. IEEE Computer Society Press, Los Alamitos, CA, 1995.

[14] E. Brown. SMTP on a High Assurance Multilevel Server. Master's thesis, Naval Postgraduate School, Monterey, CA, September 2000.

[15] S. Bryer-Joyner and S. Heller. Secure Local Area Network Services for a High-Assurance Multilevel Network. M.S. thesis, Naval Postgraduate School, Monterey, CA, March 1999.

[16] J. P. Downey and D. A. Robb. Design of a High Assurance Multilevel Mail Server (HAMMS). Master's thesis, Naval Postgraduate School, Monterey, CA, 1997.

[17] B. Eads. Developing a High Assurance Multilevel Mail Server. Master's thesis, Naval Postgraduate School, Monterey, CA, 1999.

[18] M. Gasser. *Building a Secure Computer System*. Van Nostrand Reinhold, New York, NY, 1988.

[19] J. Hackerson. Design of a Trusted Computing Base Extension for Commercial Off-The-Shelf Workstations (TCBE). Master's thesis, Naval Postgraduate School, Monterey, CA, September 1997.

[20] K. L. Heninger. Specifying Software Requirements for Complex Systems: New Techniques and their Applications. *IEEE Transactions on Software Engineering*, 2(1):2–12, January 1980.

[21] C. E. Irvine, J. P. Anderson, D. Robb, and J. Hackerson. High Assurance Multilevel Services for Off-The-Shelf Workstation Applications. In *Proceedings of the 20th National Information Systems Security Conference*, pages 421–431, Crystal City, VA, October 1998.

[22] B. Lampson. A Note on the Confinement Problem. *Communications of the A.C.M.*, 16(10):613–615, 1973.

[23] N. G. Leveson. *Safeware*. Addison Wesley, Reading, Massachusetts, 1995.

[24] Luqi. Software Evolution Through Rapid Prototyping. *IEEE Computer*, 22(5):13–25, May 1989.

[25] Luqi. A Graph Model for Software Evolution. *IEEE Transactions on Software Engineering*, 16(8):917–927, August 1990.

[26] P. Myers. *Subversion: The Neglected Aspect of Computer Security*. M.S. thesis, Naval Postgraduate School, Monterey, CA, 1980.

[27] J. Mylopoulos, L. Chung, and B. Nixon. Representing Using Nonfunctional Requirements: A Process-Oriented Approach. *IEEE Transactions on Software Engineering*, 18(6):483–497, June 1992.

[28] National Computer Security Center. *Department of Defense Trusted Computer System Evaluation Criteria*, DoD 5200.28-STD, December 1985.

[29] National Computer Security Center. *Trusted Network Interpretation of the Trusted Computer System Evaluation Criteria*, NCSC-TG-005, July 1987.

[30] National Computer Security Center. *Final Evaluation Report: Boeing Space and Defense Group, MLS LAN Secure Network Server System*, 28 August 1991.

[31] National Computer Security Center. *Final Evaluation Report of HFSI XTS-200*, CSC-EPL-92/003 C-Evaluation No. 21-92, 27 May 1992.

[32] National Computer Security Center. *Final Evaluation Report of Gemini Computers, Incorporated Gemini Trusted Network Processor, Version 1.01*, 28 June 1995.

[33] D. Parnas and J. Madey. *Science of Computer Programming*, volume 25, chapter Functional documents for computer systems, pages 41–61. October 1995.

[34] C. P. Pfleeger. *Security in Computing*. Prentice Hall, Inc., Englewood Cliffs, NJ, 2nd edition, 1986.

[35] W. W. Royce. Managing the Development of Large Software Systems: Concepts and Techniques. In *Proceedings WESCON*, August 1970.

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

[37] I. Sommerville. *Software Engineering*. Addison-Wsley, Reading, MA, Fifth edition, 1995.

[38] K. Thompson. Reflections on Trusting Trust . *Communications of the A.C.M.*, 27(8):761–763, 1984.

[39] B. Turan. Client Bootstrap Under TCBE Control. Master's thesis, Naval Postgraduate School, Monterey, CA, March 2000.

[40] A. van Lamsweerde. Requirements engineering in the year 00: A research perspective. In *Proc. ICSE'2000 - 22nd International Conference on Software Engineering*, pages 5–19, Limerick, Ireland, June 2000. ACM Press.

[41] W. H. Ware. Security Controls for Computer Systems: Report of Defense Science Board Task Force on Computer Security. Technical Report R-609-1, Rand Corporation, Santa Monica, CA, 1970.

[42] J. Wilson. Trusted Networking in a Multilevel Secure Environment. Master's thesis, Naval Postgraduate School, Monterey, CA, June 2000.

APPENDIX

A. MLS LAN SYSTEM REQUIREMENTS DOCUMENT - VERSION 0.2

A.1 Introduction

A.1.1 Purpose

The purpose of this System Requirements Document is to define the design requirements for the Naval Postgraduate School Center for InfoSec Studies and Research (CISR) Multilevel Secure Local Area Network (MLS LAN) Project.

A.1.2 Scope

This requirements document provides extensive information concerning the design requirements for each of the components of the MLS LAN project. It outlines the mandated system goals perceived for successful completion of the project and the development of an operational multilevel secure local area network. It is understood that some of the specified requirements are designated as mandatory to fulfill near-term functionality and are to be addressed in the initial design. Other requirements, where annotated, are considered to be future goals and are recorded to support long-range design specifications. This requirements document is intended to provide sufficient detail and content to assist the design team in specification definition.

A.2 System Overview

A.2.1 MLS LAN System Overview

The MLS LAN Project is an effort to provide government and commercial organizations with a cost effective, multilevel, easy-to-use office environment leveraging existing high assurance technology [Ref. 1]. The goals of the project are to produce a networking environment that provides concurrent high assurance access for network users to data at multiple sensitivity levels through the incorporation of inexpensive commercial personal computers.

The proposed systems architecture for the MLS LAN is based on the use of the Wang Government Services Incorporated XTS-300(tm) Class B3 rated server. [Ref. 2] The XTS-300 provides both mandatory and discretionary access controls, which "allow separation of users who are at different clearance levels, and prevents a lower level user from reading a higher level user's files or data". [Ref. 3] In accordance with the TCSEC Class B3 rating requirements, the XTS-300 establishes a "Trusted Computing Base" (TCB) that contains all of the Trusted Software Commands, the TCB System Services (TSS), and the Security Kernel. It is the last that implements the TCSEC defined Reference Monitor concept in the XTS-300 [Ref 4]. The MLS LAN incorporates a "logically isolated and unmistakably distinguishable" trusted communications path between the server and its clients through development of a Trusted Computing Base Extension (TCBE). The TCBE will provide a trusted network interface entity for verifiable expansion of the TCB over the communications path to the client workstation. The current hardware solution for the TCBE is to be developed using the Intel I960jx or comparable processor. The TCBE will dominate all actions of the untrusted workstation and allow connectivity into the High Assurance LAN only following the establishment of a trusted path.

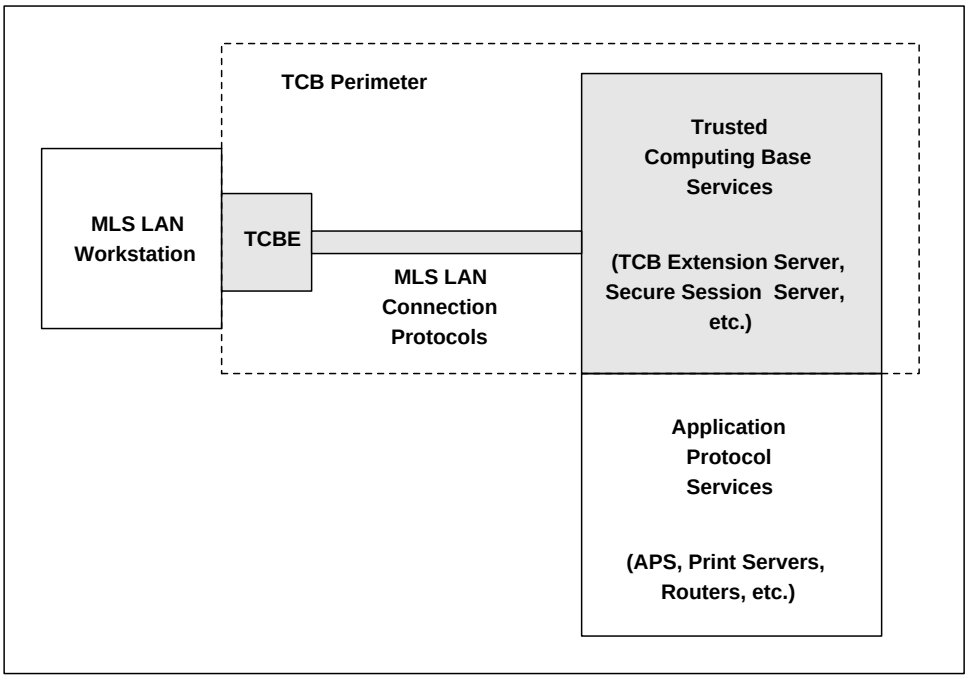

Figure 2.1 MLS LAN Component Overview

A.2.2 MLS LAN User Description

The MLS LAN user is any operator, regardless of authentication, who accesses MLS LAN resources or network functionality. A TCB Authenticated user is one who has successfully established a TCB-to-User connection and been validated by the TCB for operations within the MLS LAN. A Non-TCB Authenticated User, which is a future requirement, is one who has not been validated by the TCB. Accountability of Non-TCB Authenticated Users shall be provided using existing commercial authentication and identification mechanisms.

A.2.3 Component Descriptions

The MLS LAN is comprised of three components (Fig 2.1). The principal component is the Trusted Computing Base (TCB), which provides an fixed security perimeter for MLS LAN operations. Network functionality for access to available application software, file transfer, electronic mail, or remote printing is provided by the Network Application Protocol Services. Finally, the MLS LAN requires a workstation that acts as an agent for the User to access any required network functionality.

A.2.3.1 Trusted Computing Base

The Trusted Computing Base is an abstraction for the collection of elements of a computer system that pertain to the security policy. Its aegis encompasses all policy enforcement mechanisms, any auditing (retrieval and analysis), identification and authentication, and the interface for security administration.

A.2.3.1.1 Trusted Computing Base Services

The services provided by the MLS LAN to establish a Class B3 rated Trusted Computing Base were outlined in section 2.1 "MLS LAN System Overview". To extend this TCB securely to users additional services are required.

A.2.3.1.2 TCBE Extension Server

The use of the XTS-300 High Assurance Server enables the MLS LAN to place a trusted daemon process in the Operating System Services (OSS) Domain that can provide the protection and communications protocols necessary to establish a trusted path between the workstation and MLS LAN. This "Server" process is used to extend the TCB perimeter securely over the network to the requesting TCBE-equipped workstation. This "Server" process will provide the following functionality: user identification and authentication, session negotiation, session activation, and session termination. [Ref 6.]

A.2.3.1.3 Secure Session Server

The Secure Session Server is an additional trusted daemon "Server" process contained in the OSS. This process will only accept incoming Network Application Protocol Service requests from workstations/users that have established a session via the trusted path and the TCB Extension Server. Validated requests will be passed on to untrusted Application Protocol Servers, operating on behalf of the user, at the user's negotiated session sensitivity level [Ref 6.]

(Future Requirement) The Secure Session Server will accept Network Application Protocol Services requests from workstation/users that have not established a session, viz. Non-TCB Authenticated Users. These requests will be passed on to untrusted Application Protocol Servers, operating as a system defined anonymous user, at a system defined low secrecy, low integrity, session sensitivity level.

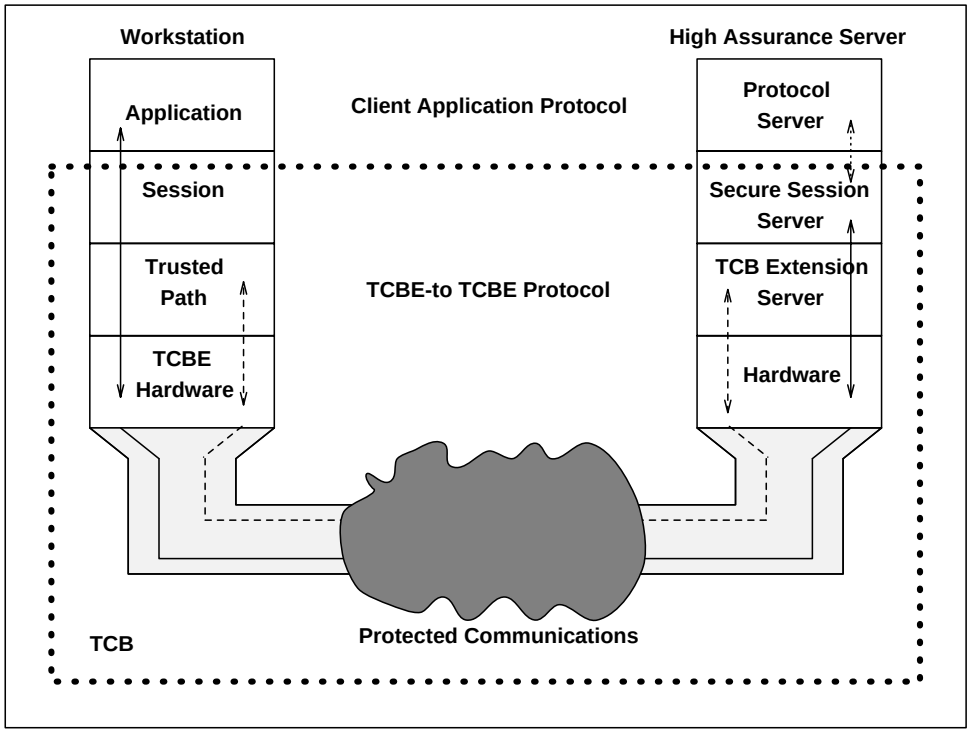

Figure 3.1 TCB Layering Abstractions

A.2.3.1.4 MLS LAN Session Database Server

The MLS LAN requires a trusted database to maintain all pertinent information concerning each unique TCB session connection. The Session Database Server must provide protection for trusted "read" functionality from all TCB entities and "write" functionality from the TCB Extension Server.

A.2.3.1.5 Trusted Computing Base Extension

The Trusted Computing Base Extension (TCBE) is a hardware-based computer subsystem that is embedded into the MLS LAN workstation. The TCBE provides the MLS LAN with a verifiable high assurance entity that can be used to extend the TCB.

A.2.3.1.6 MLS LAN Connection Protocols

The MLS LAN connection protocols define the parameters for initiation, security and communications establishment between two or more components of the MLS LAN.

A.2.3.2 Network Application Protocol Services

The MLS LAN uses the TCP/IP stack to support numerous Application Layer Protocol services such as Hyper Text Transfer Protocol (HTTP), Internet Message Access Protocol (IMAP), and File Transfer Protocol (FTP). These services are provided to the users through Application Protocol Servers (APS). While use of these application services are considered "untrusted" and external to the TCB, their access is controlled strictly through the Secure Session Server allowing access to data of multiple sensitivity levels.

A.2.4 MLS LAN

The MLS LAN workstations are the network computers employed by the user to access MLS LAN resources and network functionality.

A.3 System Requirements

A.3.1 MLS LAN Requirements

A.3.1.1

The MLS LAN shall support multiple simultaneous workstation connections.

A.3.1.2

The MLS LAN shall support simultaneous high assurance access for unique workstations operating at different sensitivity levels.

A.3.1.3

The MLS LAN shall provide access to shared resources, application protocol services, and popular application products for both TCB Authenticated Users and, in the future, Non-TCB Authenticated Users.

A.3.1.4

The MLS LAN shall provide high assurance connectivity to application protocols that give access to multiple levels of data in accordance with security policies.

A.3.2 Trusted Computing Base Requirements

This section elaborates on the requirements for the TCB in total. The overall requirements are germane to each of the

sub-components while their specific requirements are contained in subsequent sections. A abstract depiction of the MLS LAN layering is provided in Figure 3.1.

A.3.2.1 TCB Overall Requirements

A.3.2.1.1
The TCB shall provide a Secure Attention Key (SAK) mechanism to invoke a trusted path from workstations to which the TCB has been extended.

A.3.2.1.2
The TCB shall establish a trusted path communications connection between network users and the Trusted Computing Base. This trusted path shall be established for initial session authentication purposes, such as "login" or for any specified user operations that require a trusted path, such as "logout", "set session level", downgrade, change user password, etc.

A.3.2.1.3
Once the session has been established, the TCB shall not allow the TCB-to-TCBE Protocol Channel to be broken without loss of network functionality with respect to shared resources, protocol services and applications provided by the MLS LAN.

A.3.2.1.4
The TCB shall allow the user to change the current session sensitivity-level up to the configured maximum for that user.

A.3.2.1.5
The TCB shall provide assurance that the security policy will be enforced in the presence of malicious software.

A.3.2.1.6
The TCB shall provide protection against disclosure and modification of information on all communications channels used by the network.

A.3.2.1.7
The TCB shall control access all devices and networks external to the MLS LAN.

A.3.2.1.8
(Future Requirement) The TCB shall limit the allowable session sensitivity-level to the greatest lower bound between the user's clearance and the TCBE security rating.

A.3.2.2 Trusted Computing Base Extension Requirements

A.3.2.2.1
The TCBE shall support the use of Trusted Path communications with the TCB for security related operations.

A.3.2.2.2
The TCBE shall prevent data retention between session security levels and support proper object reuse.

A.3.2.2.3
The TCBE shall support a hardware mechanism that has the ability to purge all memory between session security levels.

A.3.2.2.4
The TCBE shall maintain the ability to reset the host computer system.

A.3.2.2.5
The TCBE shall support the use of a secure attention key.

A.3.2.2.6
The TCBE shall control the information flow into and out of the host computer system.

A.3.2.3 MLS LAN Connection Protocol Requirements

A.3.2.3.1
The MLS LAN shall provide a protocol that supports both the establishment of a secure interaction communications channel and the mutual authentication between two TCB entities. This protocol will be known as the "Protected Communications Channel (PCC) Protocol". This protocol will establish the security conduit through which all other MLS LAN protocols operate.

A.3.2.3.2
The MLS LAN shall provide a protocol to support communications between a TCBE equipped workstation and the TCB Extension Server. This protocol will be known as the "TCB-to-TCBE Protocol".

A.3.2.3.3
The MLS LAN shall provide a protocol to support the secure transfer of information from the TCB Extension Server to the Session Database Server to initialize or modify the data maintained on each User Session. This protocol will additionally support the query by a TCB Entity to the Session Database Server for information concerning a User Session. This protocol will be known as the Session Status Protocol.

A.3.2.3.4
The MLS LAN shall provide a protocol to support a TCBE equipped workstation connection to a MLS LAN Secure Session Server. This protocol is the conduit for application protocols and will be known as the "TCBE-to-Session Server Protocol".

A.3.2.3.5
(Future Requirement) The MLS LAN shall provide a protocol to support the connection of a workstation that is not using TCBE services to an untrusted Application Protocol Server, e.g., INTERNET or WWW.

A.3.2.3.6
(Future Requirement) The MLS LAN shall provide a protocol to support a connection of a workstation that is not using TCBE services to a MLS LAN Application Protocol Server.

A.3.3 MLS LAN Network Application Protocol Services Requirements.

A.3.3.1
The MLS LAN shall support multiple simultaneous accesses to higher layer application protocols, e.g., HTTP, IMAP or FTP.

A.3.3.2
The MLS LAN Application Protocol Servers shall provide access to shared network resources, and popular application products for TCB authenticated users.

A.3.3.3
Access to data maintained on the MLS LAN Applications Protocol Servers (APS) shall be controlled the through the TCB in accordance with the security policy.

A.3.3.4
(Future Requirement) The MLS LAN Application Protocol Servers shall provide access to shared network resources, and popular application products for Non-TCB authenticated users.

A.3.4 MLS LAN Workstation Requirements

A.3.4.1
The MLS LAN shall support the use of two configurations of inexpensive commercial personal computers:

A.3.4.1.1
Trusted Computing Base Extension (TCBE) equipped.

A.3.4.1.2
(Future Requirement) Non-TCBE equipped.

A.3.5
The MLS LAN Workstations shall support up-to-date commercial operating systems.

A.3.6
The MLS LAN TCBE Equipped Workstation shall be [in effect] "diskless thin-client" computers operating under the control of the TCBE.

A.4 MLS LAN System Restrictions

A.4.1 MLS LAN Restrictions

A.4.1.1
The MLS LAN shall support no more than one logged in user per workstation at a time.

A.4.2 Environmental Restrictions

A.4.2.1
The TCB platform shall not be subjected to physical tampering.

A.4.2.2
The TCBE hardware shall not be subjected to physical tampering.

A. Abbreviations, Acronyms, and Definitions

Abbreviations, Acronyms

APS - Application Protocol Server

CISR - Center for INFOSEC Studies and Research

FTP - File Transfer Protocol

HTTP - Hyper Text Transfer Protocol

IMAP - Internet Message Access Protocol

LAN - Local Area Network

MLS - Multilevel Secure

NPS - Naval Postgraduate School

OSS - Operating System Services

SAK - Secure Attention Key

TCB - Trusted Computing Base

TCBE - Trusted Computing Base Extension

TIC - Trusted Interaction Channel

TSS - TCB System Services

TCSEC - Trusted Computer Security Evaluation Criteria

Definitions

Trusted Computing Base: The Trusted Computing Base is defined as "The totality of protection mechanisms within a computer system - including hardware, firmware, and software - the combination of which is responsible for enforcing a security policy. A TCB consists of one or more components that together enforce a unified security policy over a product or system. The ability of a trusted computing base to correctly enforce a security policy depends solely on the mechanisms within the TCB and on the correct input by system administrative personnel of parameters (e.g., a user's clearance) related to the security policy" [Ref 4.]

Trusted Path: The Trusted Path is defined as "A mechanism by which a person at a terminal can communicate directly with the Trusted Computing Base. This mechanism can only be activated by the person or the Trusted Computing Base and cannot be imitated by untrusted software." [Ref 4.]

Session: A Session is defined as the period of interaction between a user and entities within the MLS LAN following session activation and until session termination. Sessions are established or denied based upon based on "attributes such as the location or port or access, the user's security attribute (e.g., identity, clearance level, integrity level, membership in a role), ranges of time (e.g., time-of-day, day-of-week, calendar dates) or combinations of parameters." Limitations may be placed upon user active sessions such as limitations of the number of multiple concurrent sessions or session locking based upon inactivity. [Ref 5.]

TCB Authenticated User: A TCB Authenticated user is one who has successfully established a TCB-to-User connection and been validated by the TCB for operations within the MLS LAN.

Non-TCB Authenticated User: A Non-TCB Authenticated user is one who has not been validated by the TCB.

A. References

1. Irvine, C. E., Anderson, J. P., Robb, D. A., and Hackerson, J. High Assurance Multilevel Services for Off-The-Shelf Workstation Applications. In Proceedings of the 20th National Information Systems Security Conference, pp. 421-431, Crystal City, VA, October 1998.
2. Downey, J., Robb, D. Design of a High Assurance Multilevel Secure Mail Server (HAMMS), Naval Postgraduate School, Monterey CA, September 1997.
3. XTS-300, STOP 4.4.2, Trusted Facility Manual, Document ID: FS96-371-07, Wang Government Services Inc.
4. Department of Defense Trusted Computer System Evaluation Criteria, DoD 5200.28-STD, National Computer Security Center, December 1985.
5. Common Criteria for Information Technology Security Evaluation Version 2.1, Common Criteria Project Sponsoring Organisations, August, 1999
6. Bryer-Joyner, S., Heller, S. Secure Local Area Network Services for a High Assurance Multilevel Network, Naval Postgraduate School, Monterey, CA. March 1999.