# Analysis of Terminal Server Architectures for Thin Clients in a High Assurance Network

Steven R. Balmer
39 Ibis Lane
Groton, CT 06340
srbalmer@gte.net

Cynthia E. Irvine
Department of Computer Science
Naval Postgraduate School
Monterey, CA 93943
irvine@cs.nps.navy.mil

## Abstract

*This paper examines the architectural and security impact of using commercially available, popular terminal servers to support thin clients within the context of a high assurance multilevel network. Seven potential local area network architectures were analyzed for security and utility. Three secure configurations were identified: Multiple Terminal Servers in Series; Multiple Trusted Computing Base Extension-Enhanced Terminal Servers; and Terminal Servers on a High Assurance Virtual Machine Monitor.*

**Keywords:** Multilevel Security, Thin Clients, Terminal Servers, High Assurance

## 1 Introduction:

A serious problem associated with the development of secure systems is that of object reuse. If computing devices possess storage that can be exploited by malicious software to hide sensitive information, then an accomplice executing at a lower sensitivity level can locate and reveal the information. An effective approach to object reuse must be developed for systems enforcing either identity-based or label-based policies. Both the Trusted Computer System Evaluation Criteria [19] and the Common Criteria [2] stipulate mechanisms to ensure that storage objects are voided prior to reuse.

As part of the Naval Postgraduate School (NPS) Multilevel Secure Local Area Network (MLS LAN) project [16], we have investigated object reuse in client PCs which may be used by a sequence of users who may negotiate single level sessions at any of a number of non-discretionary sensitivity levels. A project requirement is that the client PCs support popular commercial operating systems and application suites, such as Windows NT and Microsoft Office. As a consequence, high assurance object reuse is a significant challenge. At the client, storage includes, for example: RAM (system and graphics), Flash-ROM (BIOS firmware), registers, buffers, bridges, cache memory, and secondary storage (hard drive). Before reuse at a new session level or by a new user, these locations should be purged of residual information. Certain popular operating systems, e.g. Windows NT, write to their boot devices during a typical startup sequence. Thus there must be writable persistent storage at the PC–a serious concern for object reuse. Thin client computing appears to offer a solution to this aspect of the object reuse problem by relocating operating system instances to a centralized server.

This paper presents an examination of the architectural and security impact of using commercially available, popular terminal servers to support thin clients within the context of a high assurance multilevel network. Related investigations associated with our project examine other facets of the PC-based object reuse problem [3]. The remainder of this paper is organized as follows: In Section 2 the basic objectives and architecture of the NPS MLS LAN are presented along with an overview of the requirements for the trusted computing base extension to be located at client PCs. A brief review of thin client computing is given in Section 3. This is followed in Section 4 by an analysis of seven potential topologies for use of thin client terminal servers in the MLS LAN. A summary and possible avenues for future research complete the paper in Section 5.

## 2 A Multilevel Secure LAN

The NPS MLS LAN project is building a system to 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 [7].

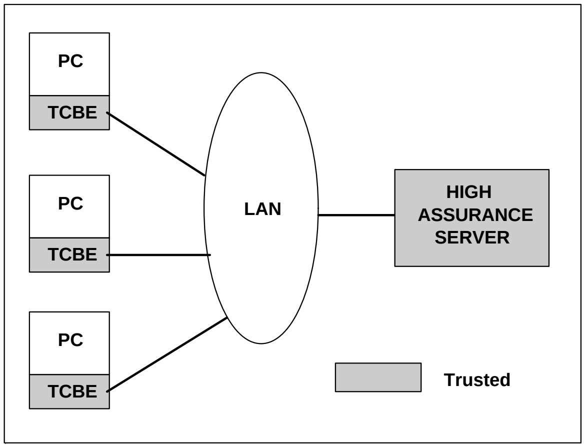

**Figure 1. NPS Multilevel Secure LAN Architecture**

The high assurance server (HAS) enforces the security policy and controls access to information. Application protocols run on the HAS and provide services and access to shared resources. Each PC is to be equipped with a Trusted computing base (TCB) 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 first major application service to be implemented in the LAN is mail. Individual components are discussed below.

### 2.1 Trusted Mail Server

The Trusted Mail Server consists of a high assurance TCB, which enforces critical security policy, and untrusted mail server instances constrained by the TCB. The server supports sharing and labeling and is functionally equivalent in terms of overall application-level protocol support to an existing mail server. Thus it is compatible with existing COTS mail client packages such as Lotus Notes, Outlook and Netscape [1].

On the high assurance platform the mail service application is instantiated within single level subjects at several access classes. The result is a multilevel mail server supporting controlled sharing of individual files. The University of Washington Internet Mail Access Protocol (IMAP) [1] mail server has been ported to the high assurance platform [10]. Adaptation of the mail server application permits the user view of the system of mailboxes at or below the user's current session level.

Both the mail spool and the mail boxes of individual users are stored at the high assurance server. We take advantage of the existing file system provided by the TCB. Initial experiments using the *Elm* user agent software provided a starting point for server organization [9]. Continuing research has allowed us to consider a variety of options for organizing mail folders.

### 2.2 High Assurance Base

The Wang XTS-300 provides our high assurance base. The principal technical consideration in choosing the high assurance base for the mail server was its ability to enforce security policy. The high assurance 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. Other factors included ease of use, available software tools, and interfaces. The principal non-technical consideration was the availability of software and hardware maintenance support.

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

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 already on the Evaluated Products List (EPL) with a Class B3 or higher digraph based upon an evaluation against the TCSEC or its network interpretation [20].

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 [7].

### 2.3 Client Workstations

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 [14, 5]. This component provides 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. The secure attention key must be available at all times.
- Protected communication channels between the TCB and the TCBE. These protected communications are based upon protocols that support both the establishment and maintenance of trusted path, and session-level communications.
- Mechanisms to ensure high assurance object reuse at the client PC. These mechanisms must address both primary and secondary storage.
- Controlled delivery of operating system and application software to client PCs. The TCBE must insure that it has control of the client and its resources at the time of boot and that control over security critical actions is maintained throughout the client session.

The TCBE is based on an add-on card and is intended to maximize compatibility with COTS products. A software-based experimentation base as well as investigations of an Intel-based prototype [25] permits initial examination of design choices.

## 3 Thin Client Computing

Long ago, all clients were thin; centralized computing facilities provided computational services to dumb terminals. For example, Multics [21] was intended to provide a centrally managed computing utility similar to typical utilities such as electric power and gas. The introduction of inexpensive workstations and PCs resulted in a paradigm shift. The desktop became autonomous and centralized services were used for network and application support services. Distributed computing took advantage of factors including: less expensive equipment, more sophisticated users, improved user interfaces, increased redundancy and heterogeneity [26]. These advantages combined with other factors produced a new set of problems including: lack of management control; imprecise synchronization; conflicting or absent standards; poorly understood configuration interaction; the inability of general-purpose desktop platforms to provide optimum performance for all tasks; and the inability to support massive applications.

The problem of system maintenance alone provides a compelling reason for reexamining a centralized computing framework [11]. In Plan 9, the creators of Unix shifted computationally intensive activities back to a high-powered central system

[22]. Newer thin client architectures relegate only display functions to the client. In some thin clients, specific protocols require the client to execute operating system or application-specific [18] graphical interfaces[8] [6], while the thinnest of clients are truly dumb, with complete screen buffers transmitted from the central computing facility [15] [24].

Thin client computing provides advantages with respect to object reuse. First, the operating system does not boot on the client. Some popular COTS operating systems, e.g. Windows NT, must write to their boot devices, which then must be volatile. This presents an object reuse problem since the boot device can be used to store salvable information. There is no guarantee that the information being written by the OS will be constant, so integrity checks such as bootstrap ratchets [4], would be infeasible. In contrast, the protocol support and graphics functions for thin clients can be stored in non-volatile memory, and volatile memory can be purged between sessions. Another advantage is centralized management of the operating system and applications, which can provide better guarantees that updates are consistent across the network.

## 4 Terminal Server Topologies

Three logical components comprise a typical terminal server architecture. The first is a multi-user server, the second is a protocol interface between the server and the client, and the last is client software that permits each PC on the LAN to act as a terminal. A terminal server delivers the graphical user interface and keyboard services to clients, while operating system and application processing takes place on the server itself.

Terminal servers provide a way to run modern COTS operating systems within the context of obsolete hardware. This is accomplished through a type of terminal emulation characterized by the capture and transmission of a computer (terminal) keyboard keystrokes and mouse events to a separate computing device (the server) for processing. The processing system then returns changes in graphics displayed on the PC's (terminal's) screen.

The topology of a small terminal server LAN is shown in Figure 2. The terminal server provides operating system and application support to a group of users with low to moderate requirements for network activity and CPU usage.

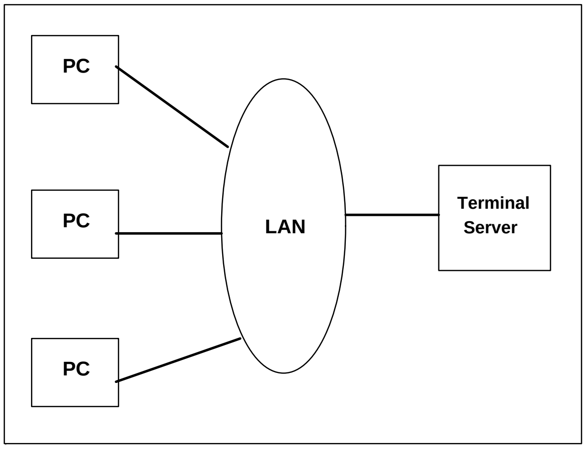

**Figure 2. Basic Terminal Server Architecture**

An enterprise-level terminal server-based LAN where users have high network and CPU usage requirements is depicted in Figure 3. Many terminal/OS servers and application servers work together to provide adequate support to users. The application server balances the load for processor-intensive applications, reducing stress on terminal server cluster processors. The question addressed in the following analysis is: Can any configuration of a multilevel secure LAN using COTS terminal

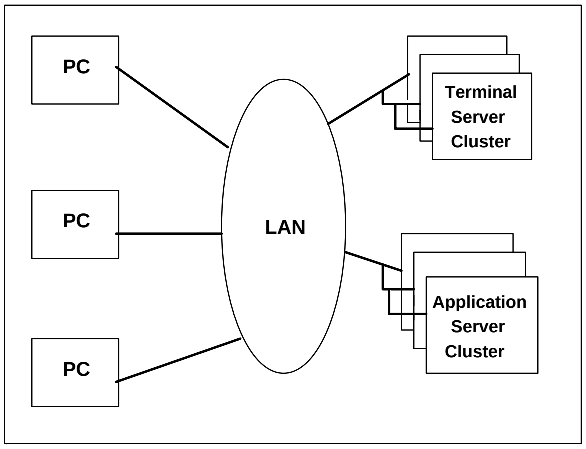

**Figure 3. Enterprise Terminal Server Architecture**

servers be secure? The configurations considered will build upon the basic MLS LAN (Figure 1). We will assume that the PCs act as if they were thin clients, without permanent, writable storage.

### 4.1 Case 1: Terminal Server as a LAN Peer

The topology in Figure 4 represents the simple addition of the terminal server to the MLS LAN. The high assurance server and the client/TCBE work as depicted in the basic LAN (Figure 1). The terminal server is added to deliver the operating system (and client applications) to the client. This network is made up of any number of PCs configured to communicate with a terminal server. Each PC is enhanced with a TCBE to support the trusted path, and encrypted communications with the high assurance server. The user would invoke the secure attention key (SAK) for authentication to the TCB and session level negotiation. Then the TCBE would permit the thin client operating system to be loaded and the user would be authenticated to the terminal server. The terminal server would then process all keyboard/mouse actions at the client. The terminal/application server could run an e-mail client at the terminal server and would access IMAP e-mail services running on the high assurance server (HAS). The IMAP server allows the user to access mail at classifications dominated by the session level negotiated with the high assurance server.

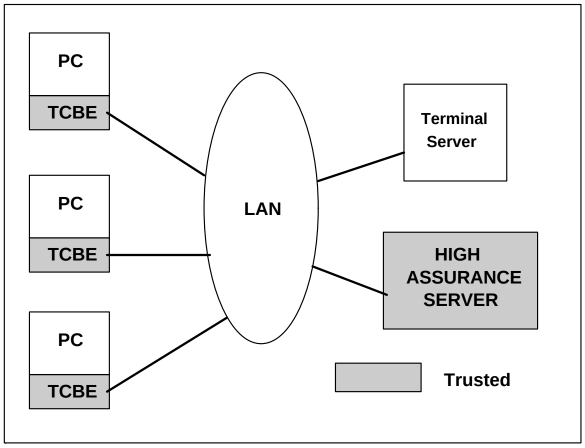

**Figure 4. Terminal Server as Peer**

**Analysis:** This configuration is efficient and scales well. The number of users on a single server could range from fifteen to forty-five depending upon usage patterns. If the expected number of PCs is large, the terminal server would expand to a cluster of single or multiple processor terminal servers running load balancing software to optimize efficiency. The terminal server on the LAN allows the terminal server to process the bulk of the network traffic without the interference of the high assurance server. Thus the high assurance server is involved only with trusted path and protocol services such as e-mail services.

In this architecture, all communications between the TCBE and the high assurance server are protected. Following identification and authentication with the HAS, communications in the network require the terminal server. There is no high assurance component at the terminal server to protect those communications.

Another vulnerability is in the terminal server itself. A typical terminal server will support, at most, no more than discretionary access controls (DAC) with low assurance. Such systems are vulnerable to malicious code. Individual users may become the victims of Trojan Horses. A “HIGH” user might unwittingly execute a Trojan Horse in the background and write information to a place accessible by a “LOW” user. Malicious code could take advantage of system flaws to acquire unrestrained access to all memory in the system. This code could grab information from another’s memory space and copy it into its memory space for exfiltration. Finally, all users must log onto the terminal server for access to supported applications. The single terminal server caches data for its own efficiency and information could be found in swap files maintained on permanent media. Current terminal servers’ capabilities can only separate user processes with low assurance, if any, and this lack of isolation is insufficient to meet the MLS LAN assurance requirements. Numerous timing and storage channels could be exploited on this unevaluated platform.

### 4.2 Case 2: Single Terminal Server in Series with High Assurance Server

Figure 5 shows a configuration in which the terminal server is attached to the high assurance server. All communications between the client and the terminal server are mediated by the high assurance server. The client and the high assurance server are attached as in the basic LAN, so the trusted path and session negotiations are as described in section 4.1. The significant difference is that the clients are not in direct communication with the terminal server. This allows all communications on the LAN to be protected through the use of high assurance mechanisms. After authenticating to the TCB, a connection to the terminal server would be created that would permit support of the thin client. The bulk of the traffic on the network would be the communications between the terminal server and the client PCs for processing keyboard/mouse events, and graphical display objects. For example, if the user wishes to access e-mail, he/she must first activate the e-mail client in the terminal server. This communications path is: COTS PC client through TCBE to high assurance server to terminal server. The application is activated on the terminal server and makes a request to the high assurance server on behalf of the user. This communications path is: terminal server to high assurance server. The high assurance server passes the information to the terminal server which is then displayed to the client. This path is: high assurance server to terminal server to HAS to client/TCBE.

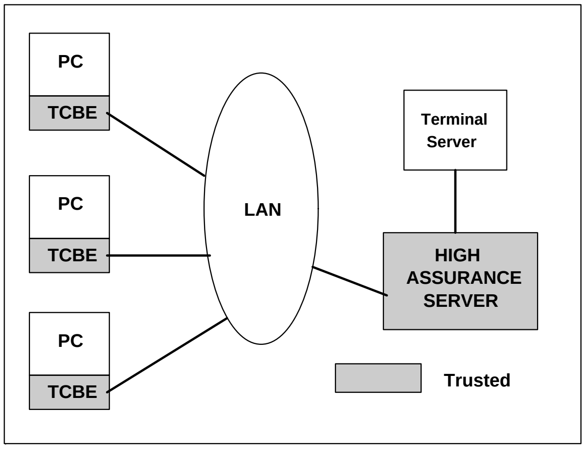

**Figure 5. Terminal Server in Series with HAS**

**Analysis:** The communications paths described above burden the HAS with significant communications processing. This configuration does not scale effectively. To handle a greater number of users and the attendant increase in traffic, additional HASs as well as the terminal servers must added to the LAN.

Security for LAN traffic is improved in this configuration as the HAS server and the TCBE at the client PC provide high assurance control over all LAN communications. There is no improvement in the security at the terminal server; it is still incapable of keeping data of different sensitivity levels separate with low, if any, assurance. There are still vulnerabilities to malicious code running at the application layer. Flaws in the operating system underlying the terminal server could be exploited.

### 4.3 Case 3: Single TCBE-Enhanced Terminal Server

Figure 6 shows a MLS LAN with an enhanced terminal server. Equipped with a terminal server TCBE that allows it to communicate securely with the client/TCBE and the HAS, the terminal server operates as a peer with the HAS.

This is very similar to Case 1. The obvious difference is the presence of a terminal server-TCBE (TS-TCBE) at the terminal server. This TS-TCBE is capable of supporting high assurance protection of network communications, and may even conduct a controlled bootstrap of the terminal server host. It differs from the client TCBEs in that it must be able to manage communications associated with each client PC logged onto the terminal server.

A typical session would begin as in Case 1. When user authentication and session level negotiation has completed, a protocol can be used to establish protected communications between the client TCBE and the terminal server TCBE. A distinct communications channel would exist for each

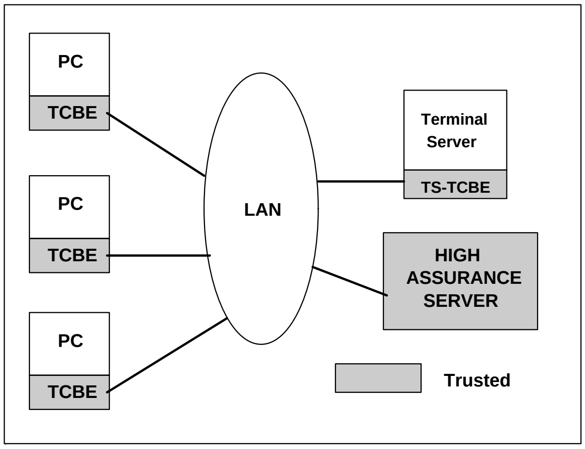

**Figure 6. TCBE Enhanced Terminal Server**

authenticated client. One or more separate protected communication channels between the terminal server TCBE and the HAS can be established. This communication channel would be used to transmit messages to the IMAP server instance on the high assurance base to and from all terminal server instances at a particular security level.

**Analysis:** This configuration allows more efficient runtime communication between the client and the terminal server because it does not involve the HAS as a pass through. It allows for protected communications on the LAN thus securing data in transit. In this respect, it is an improvement over Case 1.

This configuration does not improve security of data within the terminal server. The terminal server still has the same internal vulnerabilities, as described for Case 1.

### 4.4 Case 4: Per-Sensitivity-Level Unenhanced Terminal Servers

In this configuration, physically separated terminal servers support each classification level within the MLS LAN ( Figure 7). Each terminal server has direct access to the LAN and supports only one classification level. This system builds upon Case 1 by physically partitioning terminal servers into discrete system high domains. A user who attempts to access data at level “HIGH” and below will access the data only through the “HIGH” terminal server. Similarly a user who attempts to access data at “LOW” will only access the data through the terminal server designated “LOW”. Session startup proceeds as follows. The user at the client PC invokes the system in a fashion similar to that of Case 1 by powering on the system and/or pressing the SAK. The HAS controls identification and authentication and session level negotiation. When complete, the HAS may connect with the terminal server for the session level. The user’s PC will then complete the connection with that terminal server and begin the session.

**Analysis:** The terminal servers are not enhanced with TCBEs as in Case 3 and so have no high assurance protected communications channel between: (a) the terminal server and the Client/TCBE, and (b) the terminal server and the HAS. So, communications between the terminal server and other LAN elements are protected only by the less trusted mechanisms of the commercial product.

Physical separation between the terminal servers does not increase security; this configuration suffers the same problems as Case 1: open attacks to data in transit by malicious listeners on the LAN, and exfiltration of data within the terminal server itself. Per session data are vulnerable.

Besides its security problems, this configuration is not scalable to networks requiring a large number of sensitivity levels. In fact, more terminal

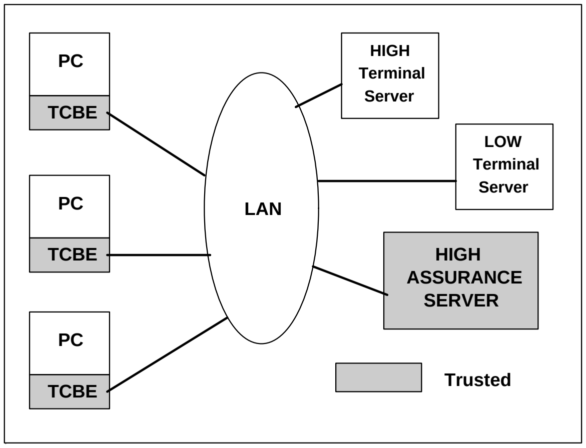

**Figure 7. Per-Sensitivity-Level Unenhanced Terminal Servers**

servers than PCs may be required in a highly compartmented operational environment.

### 4.5 Case 5: Multiple Terminal Servers in Series with High Assurance Server

The architecture in Figure 8 combines those of Cases 2 and 4. The terminal servers are connected to the high assurance server so that all communications between the clients and the terminal servers is mediated by the high assurance TCB. The connections between the terminal servers and the high assurance server are single level and the terminal servers are unable to access the LAN directly. As in Case 2, the HAS will authenticate users. Because each terminal server is running at a single level that cannot be spoofed, commercial mail clients executing on the terminal servers can request services from the IMAP server instances at the high assurance platform and can have multilevel access to mail.

**Analysis:** Security is good in this topology: requirements for multilevel access to data can be met, and the terminal servers are confined to a single security level so sensitive information cannot be leaked. This architecture suffers from the same bandwidth problems described in Case 2: all traffic between clients and the terminal servers must pass through the high assurance server. Also, the scalability problems noted in Case 4 are possible in highly compartmented environments.

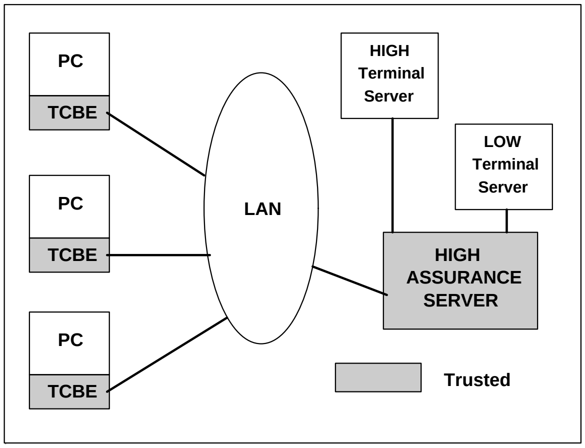

**Figure 8. Multiple Terminal Servers in Series with High Assurance Server**

### 4.6 Case 6: Multiple TCBE-Enhanced Terminal Servers

In this topology (Figure 9), TCBE front-ends are applied to each terminal server. The presence of the TCBE on each terminal server allows the terminal servers to communicate securely with the TCBEs associated with clients and with the HAS. Here successful identification, authentication, and session level negotiation between the user working through the client TCBE can be used within the LAN to provide the client with protected access to the terminal server at the appropriate session level. The TCBE at the terminal server can ensure that only those clients authorized by the HAS can avail themselves of its services. The terminal server will have a protected communication path with the HAS, so it can access HAS-managed data at all sensitivity levels permitted by network security policy.

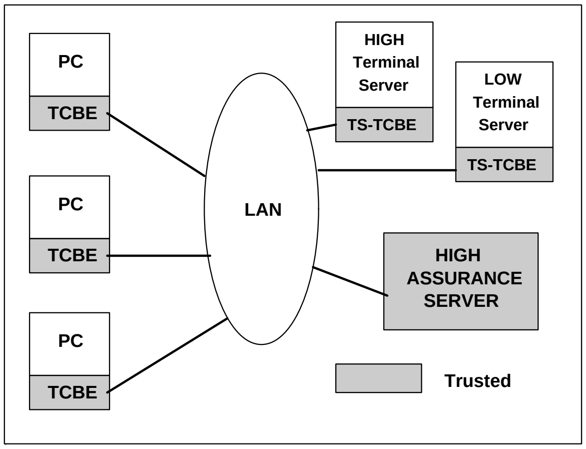

**Figure 9. Multiple TCBE-Enhanced Terminal Servers**

**Analysis:** This configuration provides high assurance of network security policy enforcement. Clients are allowed to access information at or below the negotiated session level. Each client accesses the terminal server corresponding to its current session level. Terminal servers work on behalf of clients in tandem with resident client applications and are able to successfully access only data at or below the terminal server level. The HAS enforces the multilevel security policy, managing and storing data at a range of session levels. Trusted components ensure that LAN communications are protected.

Although attacks within a closed sensitivity level are possible, there would be no compromise of nondiscretionary security policy. This approach does suffer from the same scalability problem encountered in Case 4: support of many sensitivity levels will result in many terminal servers and a proliferation of platforms.

### 4.7 Case 7: An Ideal Solution - A Secure Virtual Machine Monitor for Terminal Servers

The topology in Figure 10 represents a hypothetical LAN that uses a secure virtual machine monitor (VMM) to logically separate instances of the terminal sever. A virtual machine monitor (VMM) is capable of creating and controlling virtual replicas of a particular processor and its operating environment including address space for RAM and any permanent media required. A VMM would be able to create isolated environments for multiple operating systems or operating system instances. A secure VMM would provide assurance that information is confined within security levels. KVM provided an early example of a VMM embracing the notion of high assurance policy enforcement by a security kernel [13, 12]. The Class A1 VAX VMM is an example of such a system [17]. For the purposes of the MLS LAN, the VMM would provide a virtual machine for instances of the terminal server. For each classification level, a separate terminal server able to support multiple clients at that level could be dynamically created. To make this architecture widely available, the VMM should execute on a commodity processor. Recent analysis has demonstrated that the Intel Pentium processor is not virtualizable [23], so a different processor would be needed.

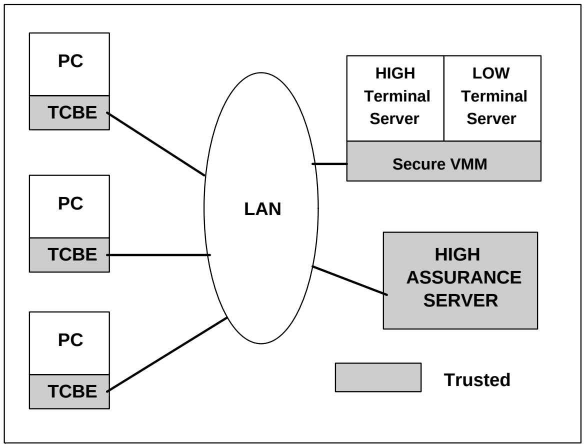

**Figure 10. Terminal Servers Running on a Secure VMM**

**Analysis:** Network security policy is supported by the three trusted elements in the LAN. The HAS enforces policy through separation of data; the secure VMM enforces policy through the creation of virtual machines at different sensitivity levels; and the TCBE supports a SAK, unspoofable interfaces, protected communications channels, and object reuse controls at the client workstation. Whether a VMM could support a large enterprise in this configuration is a matter for future analysis.

## 5 Summary

This paper has presented an analysis of seven architectures for incorporation of terminal servers into a high assurance multilevel local area network. Three secure configurations were identified: Multiple Terminal Servers in Series; Multiple TCBE-Enhanced Terminal Servers; and Terminal Servers on a High Assurance VMM. The first is likely to suffer from performance problems due to the placement of the HAS between the terminal servers and the PC-based clients. In addition, this configuration will not scale well to environments requiring many sensitivity levels. The second architecture also suffers from the same scalability problem. The VMM approach is ideal, however, at this time it is only hypothetical as no highly secure and trustworthy VMMs are available.

### 5.1 Future Work

Our analysis shows that current options for using terminal servers in a high assurance context with many sensitivity levels are impractical. For the MLS LAN project we are considering other options for provision of operating system and application clients to the PCs located on the LAN. These will be discussed elsewhere.

The hypothetical, yet from the security perspective ideal, solution merits further consideration. A high assurance VMM could dynamically support virtual machine instances at any access class. With no storage at thin clients, the danger of improper object reuse between sessions is substantially reduced and the benefits of centralized software management accrue.

Each client would need to be equipped with a TCB Extension to provide: (1) 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. (2) Protocols support for protected communication channels between the TCB and the TCBE. (3) Mechanisms to ensure high assurance object reuse at the client PC. (4) Controlled startup and delivery of software to client PCs. Recent developments in thin client computing have resulted in major simplifications and reduction of the thin client, e.g. SLIM [24], and make this approach particularly attractive. The scalability of such architectures must be examined.

## Acknowledgements

The authors wish to thank James Anderson for his encouragement and David Shifflett for his technical advice.

## References

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

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

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

[4] W. A. Arbaugh, D. Faber, and J. Smith. A Secure and Reliable Bootstrap Architecture. In *Proceedings 1997 IEEE Symposium on Security and Privacy*, page ww, Oakland, CA, May 1997.

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

[6] Boca Research, Inc. *Citrix ICA Technology Brief*, Boca Raton, FL, 1999.
http://www.bocaresearch.com
/technologies/icatech.html.

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

[8] Databeam Corporation. *A Primer on the T.120 Series Standard*, Lexington, NY, May 1997. http://www.databeam.com/standards/index.html.

[9] J. P. Downey and D. A. Robb. *Design of a High Assurance Multilevel Mail Server (HAMMS)*. M.S. thesis, Naval Postgraduate School, Monterey, CA, 1997.

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

[11] GartnerGroup. Thin-Client Desktops Cust Support Costs by 80 Percent. Datapro Reports, May 1999.
http://gartner5.gartnerweb.com/
public/static/aboutgg/pressrel/051899thin_client.html.

[12] B. Gold, R. R. Linde, and P. F. Cudney. KVM/370 in Retrospect. In *Proceedings of the 1984 IEEE Symposium on Security and Privacy*, pages 13–23, Oakland, CA, April 1984. IEEE Computer Society Press.

[13] B. Gold, R. R. Linde, M. Schaefer, and J. F. Scheid. Vm/370 security retrofit program. In *Proceedings 1977 Annual Conference*, pages 411–418, Seattle, WA, October 1977. A.C.M.

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

[15] M. Hayter and D. McAuley. The Desk Area Network. *Operating System Review*, 25(4):14–21, October 1991.

[16] 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.

[17] P. A. Karger, M. E. Zurko, D. W. Bonin, A. H. Mason, and C. E. Kahn. A VMM Security Kernel for the VAX Architecture. In *Proceedings 1990 IEEE Symposium on Research in Security and Privacy*, pages 2–19. IEEE Computer Society Press, 1990.

[18] Microsoft Corporation. *Microsoft Windows NT Server 4.0 Terminal Server Edition Resource Guide*, Seattle, WA, 1998. Penn Well Publishing Company.

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

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

[21] E. I. Organick. *The Multics System: An Examination of its Structure*. MIT Press, Cambridge, MA, 1972.

[22] R. Pike, D. Presotto, S. Dorward, B. Elandrena, K. Thompson, H. Trickey, and P. Winterbottom. Plan 9 from Bell Labs. Lucent Technologies, 1995. http://plan9.bell-labs.com/plan9/doc/9.html.

[23] J. S. Robin and C. E. Irvine. Analyzing the intel pentium's capability to support a secure virtual machine monitor. In *Proceedings of the 9th USENIX Security Symposium*, Denver, CO, August 2000.

[24] B. K. Schmidt, M. S. Lam, and J. D. Northcutt. The Interactive Performance of SLIM: a Stateless, Thin-Client Architecture. In *Proceedings 17th ACM Symposium on Operating Systems Principles*, pages 32–47, Charleston, SC, December 1999. also appears in Operating System Review33:5.

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

[26] A. Umar. *Distributed Computing and Client-Server Systems*. Prentice Hall, Inc., Englewood Cliffs, NJ, 1993.