Proceedings of the 5th IEEE Systems, Man and Cybernetics
Information Assurance Workshop, United States Military Academy,
West Point, NY, 10–11 June 2004, pp 38-45.

Overview of a High Assurance Architecture for Distributed Multilevel Security

Cynthia E. Irvine, Timothy E. Levin, Thuy D. Nguyen, David Shifflett, Jean Khosalim,
Paul C. Clark, Albert Wong, Francis Afinidad, David Bibighaus, Joseph Sears

Abstract— **A high assurance architecture is described for the protection of distributed multilevel secure computing environments from malicious code and other attacks. Component security services and mechanisms extend and interoperate with commodity PCs, commodity client software, applications, trusted components, and legacy single level networks, providing new capabilities for composing secure, distributed multilevel security. This architecture results from the realization that unless a secure system offers users comfortable and familiar interfaces for handling routine information, it will fail due to lack of user acceptability.**

I. Introduction

There is a growing need to support mandatory enforcement of confidentiality and integrity policies in hostile environments. Applicable environments include: military coalitions, responses to security emergencies at home, and business and financial relationships. Neither military computer systems and networks nor their commercial sector equivalents, are currently organized to provide high assurance support for multilevel security policy enforcement and adequate defense against increasingly sophisticated attacks. Thus we risk corruption of critical data and systems, leakage of sensitive information, and degradation of service to fundamental infrastructure systems. Industrial systems run the risk of economic espionage, while the lack of policy support for Joint Command and Control Systems constrains military operations. As shown in Table I, attacks against modern systems range from trivial to grave.

To secure mission critical information systems, new trusted computing approaches are required, involving both interoperable system security features and standardized security mechanisms. We describe an innovative high assurance architecture to provide trusted security services and integrated operating system mechanisms that can protect distributed multilevel secure computing environments from malicious code and other attacks. These security services and mechanisms extend and interoperate with existing applications and commodity clients, providing new capabilities for composing secure distributed systems using commercial off-the-shelf (COTS) components. The latter objective results from the realization that unless a secure system offers users the same comfortable and familiar interfaces used for handling routine information, it will fail due to lack of acceptability.

The Monterey Security Architecture (MYSEA) provides a trusted distributed operating environment for enforcing multilevel security policies, and utilization of support for incorporation of unmodified commodity productivity applications for user activities. It encompasses many low-assurance commercial components and relatively few specialized high-assurance elements. This arrangement permits the ongoing investment in commodity personal computer (PC) operating systems and applications to be integrated into an environment where enforcement of critical security policies is assigned to more trusted elements. Assurance is derived from the application of high assurance system design and development methods to the trusted elements as well as to the overall architecture.

The locus of policy enforcement in MYSEA is a high assurance platform, currently the DigitalNet XTS-400. We have vertically integrated application security requirements with underlying security services, and can apply an existing Quality of Security Service model and framework [1] to the integrated security structure. Additionally, MYSEA supports secure trusted path communications between the user and the trusted OS, as well as high assurance labeling for incoming traffic from legacy single level networks.

The state of the art for protecting multilevel information and for the management of security policies and security services in support of critical applications is advanced through several innovations:

- A distributed architecture for isolating trusted components in support of commercial and open source applications. The innovative use of add-on trusted components in commercial client-server systems can potentially magnify the impact of highly trusted systems.
- A trusted path mechanism for assured, unambiguous user communication with the trusted computing base, which does not depend upon client workstation security.
- Techniques for vertical integration of security policy control functions with underlying security services in a Quality of Security Service framework.

C.E. Irvine, T.E. Levin, T.D. Nguyen, D. Shifflett, J. Khosalim, P.C. Clark, A. Wong, F. Afinidad, D. Bibighaus, J. Sears: Naval Postgraduate School, Monterey, CA.

This material is based upon work supported in part by the Office of Naval Research and other government sponsors. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsors.

TABLE I

Attack Elements and System Assurance Required for Defense.

Attack Motive	Attack Strategy	Attack Resources	Threat	Assurance Required
Political-Military	Long-Term Planning	Well Funded	System Subversion	Highest
Political-Military	Mid-Term Planning	Modest to High	Malicious Code Trojan Horses	High
Malicious Amusement	Short-Term Planning	Low to Modest	Flaw Exploitation	Modest
Malicious Amusement	Ad Hoc	Low	Interface Exploitation	Low

- Secure single level connections to existing classified networks. These connections may be initiated either from clients within the multilevel network to access single level resources, or from existing single level networks to access resource on the multilevel server.

II. Monterey Security Architecture

MYSEA is a distributed client-server architecture featuring a combination of (relatively few) specialized policy enforcing components and multiple open source and commercial off-the-shelf components. The major physical components of the architecture are illustrated in Figure 1:

- High assurance MYSEA Servers which provide the locus for multilevel security policy enforcement and host various open source or commercial application protocol servers,
- Security enhanced clients comprised of commodity PCs executing popular commercial software, along with Trusted Path Extensions that provide trustworthy policy support mechanisms and thus permit distribution of server-enforced security policy across the network, and
- Existing classified single level networks connected to the high assurance multilevel servers. Complementing link encryptors, trusted components ensure proper labeling of and protection data passing back to the multilevel server.

The MYSEA Server enforces the security policy and controls access to information. A unified mandatory access control policy for both confidentiality and integrity policy is enforced by the server. It combines the Bell and La-Padula [2] and strict Biba [3] policies. The system supports read-down policies; support for regrading policies must be implemented in trusted applications that use, and are constrained by, the underlying kernel. Its core is the TCSEC [4] Class B3 evaluated DigitalNet XTS-400. (It is currently undergoing a Common Criteria [5] evaluation.) We have augmented the existing TCB with services to support high assurance remote client authentication, session management, and connection to legacy single level networks. A suite of application protocol services, including SMTP, IMAP, and HTTP, has been ported to the multilevel server. These application protocol servers provide services and interfaces to shared resources along with multilevel views of information to clients. When the augmented TCB is combined with untrusted, but policy constrained (and, in some instances, policy aware) application protocol servers, the result is the MYSEA Server. MYSEA clients are PCs equipped with a Trusted Path Extension device that provides local MYSEA policy support. Trusted Channel Modules provide high assurance labeling of information entering the multilevel server from legacy single level networks. The MYSEA Server(s) communicate only with each other, with Trusted Path Extension(s) (TPE), or with Trusted Channel Module(s) (TCM) or single level networks with static sensitivity designations. Other components connected to the network will be ignored. (Note: encryption devices may also be added to the network.) Multiple MYSEA Servers provide scalability within the desired security policy perimeter.

A. MYSEA Concept of Operation

Using the Trusted Path Extension at the PC allows users to log on to the MYSEA system by way of a trusted path. This establishes an identity for audit and access control purposes, and then establishes session security attributes such as current session level. Subsequently, the user can log on to the native client OS at the PC and use (1) standard commercial client software (e.g., web browser or e-mail program) to access applications supported by the MYSEA Server or by servers on a connected single level existing network, or (2) use any applications on the local PC. From the PC the user can access any level of server data allowed by the security policy (for example, reading domains of data that are lower in sensitivity than the negotiated session level) as well as access locally created data. By again invoking the trusted path, the user can request to modify session security attributes, such as *session level.* During such negotiations, the Trusted Path Extension ensures that client access to the network is blocked.

B. MYSEA Components

MYSEA consists of the following component hierarchy:

- MYSEA Server
  - Policy-aware application protocol servers
  - High Assurance Multilevel Platform
    * trusted path services
    * security support service
    * secure session services
    * quality of security services (QoSS)

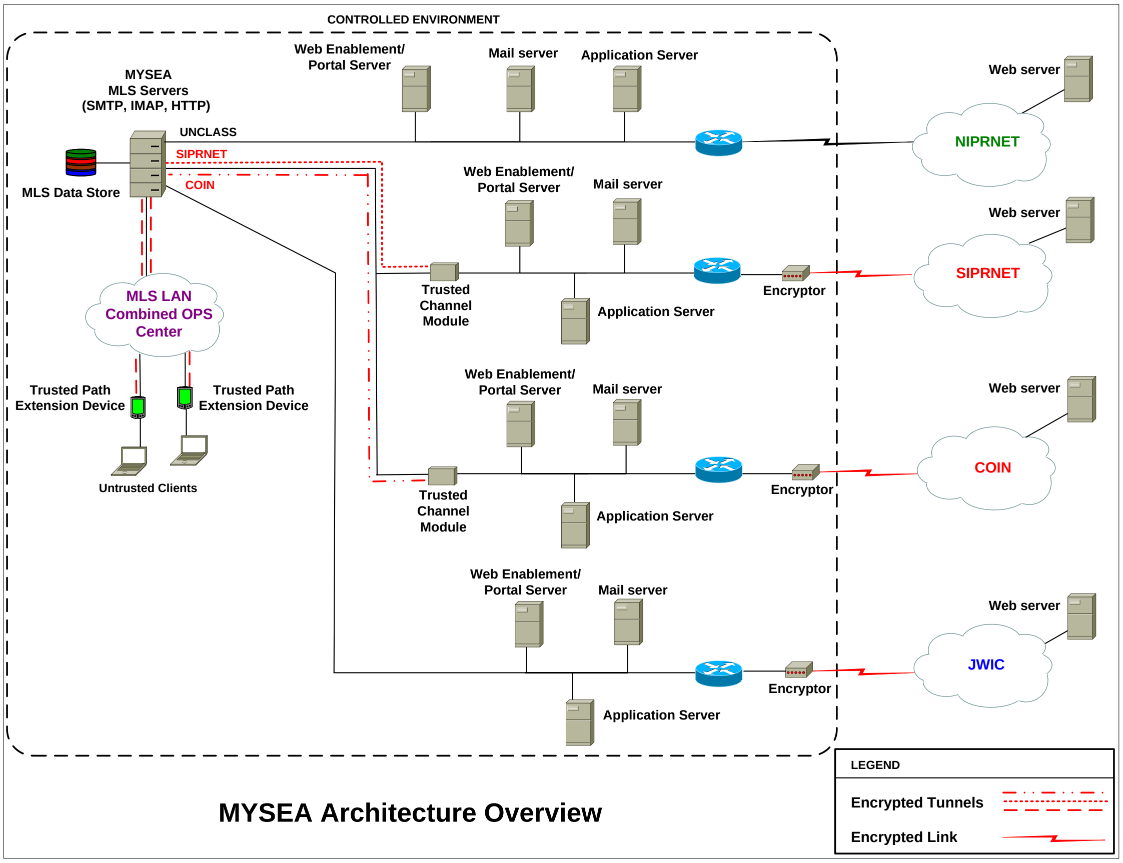

Fig. 1. Distributed Multilevel Secure Architecture.

    * cryptographic services
    * multilevel security kernel
- MYSEA Client
  - Trusted Path Extension
  - COTS PC, including unmodified:
    * operating system
    * user interface
    * applications
    * network connections
- Existing Single Level Networks
  - pre-existing single level servers and clients
  - high quality encryption to the MYSEA Server
  - Trusted Channel Module

B.1 MYSEA Server

Each MYSEA Server consists of the DigitalNet STOP operating system [6], which enforces critical multilevel security policies, multilevel services to support distributed high assurance, and assorted untrusted application server instances. Constrained by the policy enforcement mechanisms of the underlying security kernel, application servers play no role in mandatory policy enforcement, and are functionally equivalent in terms of overall application-level protocol support to a commodity application server for the particular protocol provided. Thus, each application server is compatible with existing commodity client packages. Additionally, information managed by application servers can be organized to support such sharing as allowed by the kernel policy, as well as advisory labeling.

B.2 Server TCB

The foundation for the server TCB (depicted in Figure 2) is the DigitalNet STOP security kernel. The STOP kernel creates labeled protection domains and associates security attributes with active and passive entities exported at its interface. The DigitalNet system provides multilevel secure file system support, which provides for the global and persistent separation of data into its respective domains. Other security services that have been integrated into the trusted system are described below.

B.3 Trusted Path Services

Native XTS-400 trusted path support for local terminals has been supplemented with trusted path services for remote MYSEA clients. Trusted Path Services maintains the state of the user-to-MYSEA interaction; for example, a user may be logged in with default security attributes, but may not have started a session executing untrusted application code. For remote entities, Trusted Path Services provides an interface to the Security Support Services component to support identification and authentication, negotiation of domain or domain range, password modification, account creation and deletion, and user security attribute maintenance. Once a session has been established, the Trusted Path Services provides a distributed Session Status Database to the Secure Session Services component.

B.4 Secure Session Services

The Secure Session Services (SSS) component is used to launch instances of untrusted, constrained application protocol servers and remote client-side applications. It provides trusted policy-sensitive services, with functionality similar to that of classic inetd implementations and supports standard application protocol transmissions. The SSS accesses the Session Status Database, maintained by the Trusted Path component, to determine the security attributes to associate with each application protocol server.

This Session Status Database contains tuples that uniquely identify the user, the client associated with the user, the status of the user session, the security attributes of the session, and other security relevant information. Through a session status communication mechanism, information in the Session Status Database can be provided to distributed multi-policy platforms, thus providing a single sign-on and session level capability.

B.5 Quality of Security Service Support

For dynamic management of its security and performance characteristics, the QoSS Manager is the external QoSS interface to MYSEA. It governs security and performance factors of the various MYSEA components. The QoSS security and connectivity database is managed by the QoSS manager on the MYSEA server, and is distributed to the TPEs and TCMs, as needed.

The QoSS manager provides a user interface so that decision makers can set the security posture of the network. This simple interface hides the underlying complexity of the QoSS mechanisms [7].

B.6 Constrained Application Protocol Servers

The Secure Session Server provides instances of standard protocol servers for each client or for equivalence classes of clients. The Session Status Database is used to assign security attributes to protocol servers launched on behalf of requesting clients. Thus, protocol servers are assigned security levels reflecting the policy enforced by the underlying security kernel.

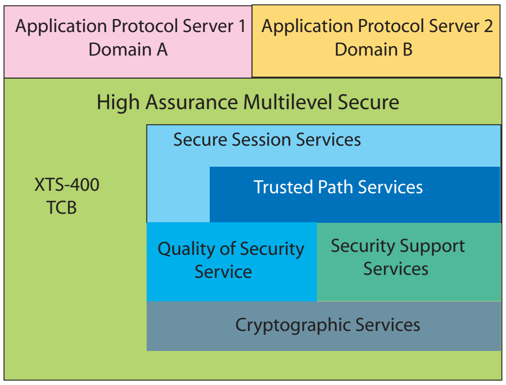

Fig. 2. MYSEA Server.

Protocol servers can take two forms. The first is a standard, policy-unaware protocol server restricted to accessing files and other objects associated only with the current session level. The second type is policy-aware, e,g, a file system, [8] and is able to take advantage of security policy domain relations that permit limited modes of access to certain other domains (e.g., “read down” for mandatory confidentiality policies).

Among the policy-aware application servers adapted to the MYSEA environment are: Internet Mail Access Protocol (IMAP) based on the University of Washington IMAP server [9], Hypertext Transfer Protocol (HTTP) based on Apache [10], and Simple Mail Transfer Protocol (SMTP) [11]. Little or no code modification was needed for adaptation to the multilevel environment.

Other protocol servers can be more tightly integrated with the underlying TCB. For example the Network File System (NFS) [12] can be implemented either as an application or in an operating system layer above the security kernel.

B.7 MYSEA Clients

MYSEA clients consist of two physical components: a Trusted Path Extension and an untrusted personal computer (see Figure 3). The PCs are typical COTS products hosting a popular commercial operating system and a commercial application suite or a thin client that accesses remote applications. The application suite includes client software intended to access standard application protocol servers. For example, mail service clients might include: Lotus Notes, Outlook, or Netscape [11].

When a user chooses to change security-policy domains, certain policies require that information associated with the previous domain be purged from the untrusted PC, e.g. previous session information cannot be reused by subsequent sessions in conflict with the distributed security policy. To ensure that these object reuse requirements are met, clients are *diskless*, with sufficient volatile RAM-disk

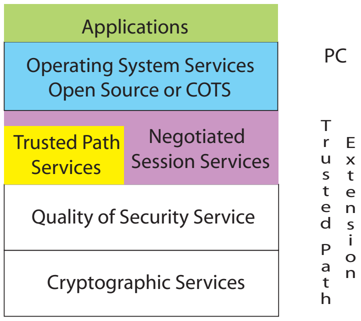

Fig. 3. MYSEA Client.

capability to support a wide variety of user applications. We have created a configuration of Windows XPEmbedded that we term *stateless-professional*. It can be booted from a non-writeable source into RAM. User preferences are to be stored on the MYSEA server. Thus a user may configure different preferences for different session levels and could have, for example, desktop advisory indicators of the current session level.

B.8 Trusted Path Extension

Not only do trusted elements of the MYSEA system provide runtime policy enforcement, they also provide services for the enforcement of supporting policies. To create a distributed TCB, the architecture includes a Trusted Path Extension (TPE) at each client.

The TPE maintains its own self-protecting domain separate from the user and PC domains. The use of a separate processor for the TPE ensures that it cannot be subverted by malicious software on the PC. Architecturally, the TPE provides the PCs only access to the network.

Currently, the TPE has a handheld form factor. The Trusted Path Extension performs IP network address translation for all IP traffic going between the PC and the LAN. User trusted path I/O occurs via the handhelds native keyboard and screen.

Simplicity has been a primary design goal for the TPE. The objective was not to construct a second operating system for the PC; it does not require the complexity and rich set of services provided for a general purpose system. We are creating a high assurance separation kernel [13] to provide a minimal set of services for the TPE.

The Trusted Path Extension can be viewed as a minimized system functioning as a *drone* in response to commands from the MYSEA server for controlling the PC and managing I/O with the user. The TPE, under MYSEA server direction, supports the following services:

Secure Attention Key. This service permits users to initiate unambiguous communication with the XTS-400 trusted path services for unspoofable presentation and capture of security-critical data. The secure attention key causes a TPE state change such that an unforgeable communications path between the user and the security functions of the server (viz. a trusted path) is established.

Trusted Path Services. When the trusted path is invoked, the user may elect to input security critical information, such as a password. The trusted path services ensure that prompts from the server are displayed and that an input mechanism for replies is available.

Controlled LAN Access. Provide non-bypassable, controlled access to the LAN from the PC. Malicious software on the PC cannot bypass the TPE.

Communications and Cryptographic Services. Provide protected communication channels between the server and the TPE. These protected communications are based upon protocols that support both the establishment and maintenance of a trusted path and session-level communications, such as to initiate communication with the server (via the secure attention key), as well as to receive and to respond to commands from the MYSEA Server.

Quality of Security Service (QoSS). Complex and adaptive networks may require *security on demand*. When conditions on the network change, requirements for security may also change. In response to a change notification, QoSS mechanisms located on the TPE can modify the protection services afforded an ongoing session. The selection of protection mechanisms for client-server communications may be based upon network conditions such as INFOCON mode. A version of IPSec adapted to provide automated, dynamic QoSS through the use of an enhanced version of a policy server such as Keynote [14] permits selection of protection mechanisms for MYSEA Servers.

B.9 Trusted Channel Module

Similar to the TPE, which provides a secure unforgeable connection between the user and the MYSEA Server, the TCMs primary function is to provide a secure unforgeable communication channel between a single level network and one or more MYSEA Servers. Once the TCM successfully authenticates with a MYSEA Server, all data arriving at the MYSEA Server from that network is properly labeled by the MYSEA server. This allows applications to run unmodified, thus enabling users to use familiar COTS applications. If required by the overall system security policy, high assurance encryption devices can be placed at various points in the MYSEA Architecture.

III. MYSEA Developmental Assurance

Our rigorous security engineering and development process [15] is intended to support high assurance evaluation of trusted components. Development begins with the capture of the threat model and the security policy to be enforced and an interpretation of that policy in terms of an abstract computer system. This results in a formal security policy model and subsequent evidence that policy enforcement objectives are met. In concert with the formal activities, the engineering team develops a series of specifications that ranges from threat model and high level requirements to detailed implementation documents and code. The system requirements specification incorporates security in conjunction with all other requirements.

Starting with a threat model and a system requirements specification (see [15]), we developed a system architecture. From these, we derive functional specifications and a corresponding detailed design specification for specific components. Concurrent development of requirements, functional, and design specification allow generalizable notions to be identified and abstracted for inclusion in the higher-level documents. Conversely, detailed items more appropriate for the lower-level specification can be moved down. All development undergoes analysis and testing.

IV. Related Work

The research defined in this paper builds on a variety of previous efforts. It extends work to construct a multilevel secure LAN [16], [17], [18], [19], [20], [9], [11], [21], [22], [23], [24]. This previous project resulted in development of prototype low assurance networking modules to support the following functions: (1) a trusted path between clients and the server, (2) session-level negotiation at the server from the clients, and (3) secure single-level session communications on the Ethernet for clients at different session levels (i.e., different domains communicate with the server through a single physical network device).

A. *User Access to Multilevel Secure Data via Commercial Workstations and Applications*

Hinke suggested the notion of a high assurance server to provide a locus of multilevel secure control to single level clients [25]. In that design sketch, clients were relegated to a single level and were connected to the multilevel server via single level network links. Although possibly useful in certain static situations, the architecture does not provide the flexibility inherent in the MYSEA design. By restricting the client to a single level throughout its lifetime, users must access multiple clients in order to manipulate information at several levels. In contrast, MYSEA allows clients to renegotiate session levels and users need only one client.

Rushby and Randell [26] describe a design for a distributed secure system that utilizes trusted network interface units (TNIUs) to connect workstations at different access classes to a local area network, through which access to a distributed multilevel file server is provided. Identification and authentication of users, as well as session level negotiation via the TNIUs is also described. Over and above this functionality, the MYSEA architecture also allows a more general purpose client-server operating environment, whereby new application servers can be easily added to the system, and thin clients are easily supported.

Various *virtual machine monitor* approaches have been suggested [27], [28], [29] for supporting COTS applications while reliably separating different domains of data. In general, for these approaches to be trustworthy requires both the use of strictly virtualizable hardware [30], and a trustworthy monitor mechanism for separating the activities of the virtual machines. Creating a monitor sufficiently trusted to both separate different domains of activity, and allow read-down to less sensitive domains (as does MYSEA) is all the more difficult. While at least one was designed to provide high assurance read-down capabilities [28], it was never fielded. The VMM approach remains problematic for separation of different domains of data because many current microprocessors are not strictly virtualizable [31], leading to complex software solutions, and because of the difficulty of creating a trusted monitor.

Non-distributed approaches to support access to multilevel data via COTS applications have been proposed in Seaview [32], [33], Purple Penelope [34], and some VMM architectures (see above). In each of these approaches, a separate process is created for each security level. Purple Pennelope has limited assurance, as it runs as a user-level application wrapping Windows NT, and it does not support a modifiable session level. The others rely on an underlying reference validation mechanism that controls access to multilevel data. The MYSEA project extends certain concepts from these projects into a distributed environment.

Replication architectures [35] provide a simple technique to achieve near-term multilevel security by copying all information at low security levels to all dominating levels. On a small scale, they may work rather well; on a large scale, in terms of both numbers of documents to be replicated and numbers of security levels to be replicated to, they are problematic. The preponderance of DoD information is either unclassified or designated sensitive but unclassified. Similar proportions hold in the commercial sector. Replication of vast amounts of data to all higher levels seems infeasible. MYSEA does not use replication as a fundamental mechanism, so it avoids these problems.

The Naval Research Laboratory (NRL) Network Pump [36] was developed to allow messages from a low sensitivity level to be sent to a high sensitivity level, and to prohibit messages and other information from going in the reverse direction. Additionally, the NRL Pump has been proposed as part of an overall network architecture to provide a more general two-way connectivity between multiple subnets at different security levels, resulting in a multiple single-level (MSL) network [37]. Here, information is also processed by an automated filter-guard to allow policy-approved flows from higher to lower domains. The MSL network approach has several drawbacks that the MYSEA avoids:

- The capital and administrative cost of separately maintained local area networks (LANs)
- The decidability challenge when attempting to provide

an automatic and reliable information filtering mechanism

- The cost of filter rule maintenance for changing policies
- The technical challenge of filtering complex information structures, such as multimedia.

Starlight [38] was designed to support logically separate single-level workstations connected by a switch to data management subsystems at different (single) levels. Software associated with the switch ensures that the current level of the workstation matches the level of data subsystem indicated by the switch setting. Starlight also allows low confidentiality information to flow through the switch to high sessions, providing a "read-down" capability. This approach has the same basic drawbacks as the MSL network, described above.

The Novell Trusted Workstation Partnership [39] defined an architecture for separating clients in different security domains with their Class C2 evaluated network software. An instantiation of this approach was developed to separate the different file system domains, but, neither the products nor detailed documentation are available.

B. Other Multilevel Variations

The *rule set based access control* (RSBAC) system [40] is a Linux extension wherein all security relevant system calls are routed through a central decision component. Access-control decisions are based on the type of access and on attributes attached to the calling subject and to the target to be accessed. RSBAC is not high assurance, has an incomplete policy for network connections and lacks the functionality of even Class B1 of the earlier TCSEC.

The Security-Enhanced (SE) Linux project is an approach to controlling multiple information domains in an open source operating system [41], [42]. The Security-Enhanced Linux project has not yet defined several mechanisms provided by MYSEA:

- Remote-client login to the trusted OS
- Trusted path communications with the trusted OS
- Changing a user session security level
- A mechanism for assigning security-domain context to a newly received network connection
- Trusted, rather than client, support for IPsec message labeling
- Support for untrusted clients, i.e., clients not based on Security-Enhanced Linux.

Content-based Information Security [43] relies on various authentication and cryptographic technologies to mediate user's access to information, but provides no underlying basis of trust to ensure against subversion or malicious software that might corrupt or leak information.

C. Trusted Path

Trusted path refers to mechanisms that provide assurance that security-critical functions are provided by the **real** system rather than masquerading software. Commercial systems, such as Windows [44], Trusted Solaris [45], and XTS-400 [46] have implemented trusted path mechanisms. In the case of Windows and Solaris, it is notable that the processing of security requests is handled, at least partially, outside of the system security perimeter (unless the entire system is included within that perimeter, thus nullifying any possible assurance arguments). In contrast to the MYSEA architecture trusted path mechanism, the XTS-400 does not support a remote trusted path.

V. Conclusion

The Monterey Security Architecture (MYSEA) provides a trusted distributed environment for enforcing multilevel security policies, and supports unmodified COTS productivity applications. The architecture encompasses a combination of many (untrusted) commercial components and relatively few trusted multilevel secure elements.

MYSEA introduces several innovations for protecting multilevel data and for managing security policies and security services in support of critical applications, including:

- A distributed high assurance multilevel architecture that utilizes commercial and open source applications.
- A trusted path mechanism.
- Techniques for vertical integration of security policy control functions with underlying security services.
- Access to existing single-level networks.

Our future plans include additions and enhancements to MYSEA. With a user-level port of the Network File System (NFS), we intend to develop a more privileged non-kernel domain version of NFS on the XTS-400. We are exploring multilevel SMB [47] services. We are extending our QoSS framework to the multilevel environment. We are investigating single sign-on support for simplified access to legacy systems and policy-enhanced remote login for users on existing single-level networks without the TPE.

Our Trusted Computing Exemplar (TCX) Project [13] complements MYSEA. The TCX kernel will be evaluatable at EAL7 under the Common Criteria. The high assurance Trusted Path Extension and Trusted Channel Module will be used as early examples of the TCX kernel.

References

[1] C. E. Irvine and T. E. Levin, "Quality of Security Service," in *Proceedings of the New Security Paradigms Workshop*, (Balleycotten, Ireland), pp. 91–99, ACM Press, September 2000.

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

[3] K. J. Biba, "Integrity Considerations for Secure Computer Systems," Tech. Rep. ESD-TR-76-372, MITRE Corp., 1977.

[4] *Department of Defense Trusted Computer System Evaluation Criteria*. No. DoD 5200.28-STD, National Computer Security Center, December 1985.

[5] *ISO/IEC 15408 - Common Criteria for Information Technology Security Evaluation*. No. CCIB-99-031, International Organization for Standardisation, version 2.0 ed., August 1999.

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

[7] R. Mohan, T. E. Levin, and C. E. Irvine, "An Editor for Adpative XML-Based Policy Management of IPsec," in *Proceedings of the 19th Computer Security Applications Conference*, (Las Vegas, NV), pp. 276–285, IEEE Computer Society, December 2003.

[8] C. E. Irvine, T. Acheson, and M. F. Thompson, "Building Trust into a Multilevel File System," in *Proceedings 13th National Computer Security Conference*, (Washington, DC), pp. 450–459, October 1990.

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

[10] 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.

[11] T. Everette, "Enhancement of Internet Message Access Protocol for User-Friendly Multilevel Mail Management," Master's thesis, Naval Postgraduate School, Monterey, CA, September 2000.

[12] R. Sandberg, D. Goldberg, S. Kleiman, D. Walsh, and B. Lyon, "The Design and Implementation of the Sun Network File System," in *Proceedings of the USENIX Conference*, (Portland, OR), pp. 119–130, 1985.

[13] C. E. Irvine, T. E. Levin, T. D. Nguyen, and G. W. Dinolt, "The Trusted Computing Exemplar Project," in *Proceedings of the Workshop on Information Assurance and Security (to appear)*, (West Point, NY), June 2004.

[14] M. Blaze, J. Feigenbaum, and A. D. Keromytis, "KeyNote: Trust Management for Public-Key Infrastructures," in *Proceedings of the 1998 Security Protocols International Workshop*, (Cambridge, England), pp. 59–63, Springer LNCS vol. 1550, April 1998.

[15] C. E. Irvine, T. Levin, J. D. Wilson, D. Shifflett, and B. Pereira, "An Approach to Security Requirements Engineering for a High Assurance System," *Requirements Engineering*, vol. 7, no. 4, pp. 192–208, 2002.

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

[17] S. Bartram, "Supporting a Trusted Path for the Linux Operating System," Master's thesis, Naval Postgraduate School, Monterey, CA, June 2000.

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

[19] 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.

[20] P. C. Clark, "Supporting Mandatory Access Control in an Educational Environment," in *Proceeding of the 23rd National Information Systems Security Conference*, (Arlington, VA), pp. 418–429, October 2000.

[21] 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.

[22] 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*, (Crystal City, VA), pp. 421–431, October 1998.

[23] R. K. Rossetti, "A Mail File Administration Tool for a Multilevel High Assurance LAN," Master's thesis, Naval Postgraduate School, Monterey, CA, September 2000.

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

[25] T. Hinke, "The Trusted Approach to Multilevel Security," in *Proceedings of the Computer Security Applications Conference*, pp. 335–341, December 1990.

[26] J. Rushby and B. Randell, "A Distributed Secure System," in *Computer*, pp. 55–67, May 1983.

[27] T. Borden, J. Hennessy, and J. Rymarczyk, "Multiple Operating Systems On One Processor Complex," *IBM Systems Journal*, vol. 28, no. 1, pp. 104–123, 1989.

[28] 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 of the IEEE Symposium on Research on Security and Privacy*, pp. 2–19, IEEE Computer Society Press.

[29] S. R. Balmer and C. E. Irvine, "Analysis of Terminal Server Architectures for Thin Clients in a High Assurance Network," in *Proceedings of the National Information Systems Security Conference*, (Baltimore, MD), pp. 192–202, October 2000.

[30] R. Goldberg, *Architectural Principles for Virtual Computer Systems.* Ph.D. thesis, Harvard University, Cambridge, MA, 1972.

[31] 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.

[32] D. E. Denning, T. F. Lunt, R. R. Schell, W. Shockley, and M. Heckman, "Security Policy and Interpretation for a Class A1 Multilevel Secure Relational Database System," in *Proceedings 1988 IEEE Symposium on Security and Privacy*, (Oakland, CA), IEEE Computer Society Press, April 1988.

[33] T. F. Lunt, R. R. Schell, W. Shockley, M. Heckman, and D. Warren, "A Near-Term Design for the SeaView Multilevel Database System," in *Proceedings 1988 IEEE Symposium on Security and Privacy*, (Oakland), pp. 234–244, IEEE Computer Society Press, 1988.

[34] B. Pomeroy and S. Weisman, "Private Desktops and Shared Store," in *Proceedings of the 14th Computer Security Applications Conference*, (Phoenix, AZ), pp. 190–200, IEEE Computer Society, December 1998.

[35] J. Froscher, M. Kang, J. Mcdermott, O. Costich, and C. E. Landwehr, "A Practical Approach to High Assurance Multilevel Secure Computing Service," in *Proceedings 10th Computer Security Applications Conference*, (Orlando, FL), pp. 2–11, December 1994.

[36] M. H. Kang, J. N. Froscher, and B. J. Eppinger, "Towards and Infrastructure for MLS Distributed Computing," in *Proceedings of the Fourteenth Annual Computer Security Applications Conference*, (Phoenix, AZ), pp. 91–100, IEEE Computer Society Press, December 1998.

[37] M. H. Kang and I. Moskowitz, "Design and Assurance Strategy for the NRL Pump," *IEEE Computer*, vol. 31, pp. 56–64, April 1998.

[38] M. Anderson, C. North, J. Griffin, R. Milner, J. Yesberg, and K. Yiu, "Starlight: Interactive Link,"

[39] J. Epstein, G. Grossman, and R. Schell, "Component Architectures for Trusted Netware," in *Proceedings of the 18th National Information Systems Security Conference*, vol. 2, (Baltimore, MD), pp. 455–463, October 1995.

[40] A. Ott, "The Rule Sett Based Access Control (RSBAC) Linux Kernel Security Extension," in *8th International Linux Kongress*, (Enschede, Netherlands), Linux-Kongress, November 2001.

[41] P. Loscocco and S. Smalley. http://www.nsa.gov/selinux/slinux-abs.html, October 2000.

[42] S. Smalley and T. Fraser, "A Security Policy Configuration for Security-Enhanced Linux," tech. rep., NAI Labs, January 2001.

[43] C. Sanders, "Information Support to Multinational Operations," *The Edge*, vol. 5, July 2001.

[44] Microsoft, "Windows 2000 Evaluated Configuration Administrator's Guide, Version 1.0," tech. rep., Microsoft Corporation, Redmond, WA, 2002.

[45] Sun Microsystems, Palo Alto, CA, *Trusted Solaris Security Features Users Guide*, 1994.

[46] Wang Government Services, Inc., McLean, VA, *XTS-300 User's Manual, Document ID: FS92-373-07*, March 1998.

[47] R. Sharpe, "Just What is SMB?," *http://samba.anu.edu.au/cifs/docs/what-is-smb.html*, October 2002.