MYSEA: The Monterey Security Architecture

Cynthia E. Irvine, Thuy D. Nguyen, David J. Shifflett, Timothy E. Levin,
Jean Khosalim, Charles Prince, Paul C. Clark and Mark Gondree
Department of Computer Science, Naval Postgraduate School
Monterey, California 93943
irvine{tdnguyen, shifflett, levin, jkhosali, cdprince, pcclark, mgondree}@nps.edu

ABSTRACT

Mandated requirements to share information across different sensitivity domains necessitate the design of distributed architectures to enforce information flow policies while providing protection from malicious code and attacks devised by highly motivated adversaries. The MYSEA architecture uses component security services and mechanisms to extend and inter-operate with commodity PCs, commodity client software, applications, trusted components, and legacy single level networks, providing new capabilities for composing secure, distributed multilevel secure solutions. This results in an architecture that meets two compelling requirements: first, that users have a familiar work environment, and, second, that critical mandatory security policies are enforced.

Categories and Subject Descriptors: D.4.6 **Software**: Operating Systems – *Security and Protection, Organization and Design*

General Terms: Design; Security

Keywords: access controls, authentication, information flow controls, cryptographic controls

1. INTRODUCTION

Governments and organizations call for the enforcement of mandatory confidentiality and integrity policies, yet these same policies now mandate the selective sharing of information among individuals and groups with differing sensitivity attributes [32, 1]. Applicable environments include: military coalitions, agencies and organizations responding to security emergencies, and mandated sharing in 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. The lack of robust security risks 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 intelligence and Joint Command and Control Systems constrains government and military operations.

STC'09, November 13, 2009, Chicago, Illinois, USA.

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 inter-operate 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) [37, 34, 39, 38, 36, 55, 54] provides a trusted distributed operating environment for enforcing multilevel security policies. Careful design allows it to encompass many low assurance commercial components and commodity productivity applications, with relatively few specialized high-assurance elements. This arrangement protects an organization's ongoing investment in commodity desktop systems and applications, and permits these components to be integrated into an environment where enforcement of critical security policies is assigned to more trusted elements. Trust 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 federation of high assurance servers. We have vertically integrated application security requirements with underlying security services, and can apply an existing Quality of Security Service model and framework [47] 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:

1. A *distributed security architecture* incorporating trusted components in support of multilevel information processing using commercial and open source applications. This innovative use of trusted components in a client-server architecture significantly leverages the impact of highly trusted systems.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
NOV 2009		**00-00-2009 to 00-00-2009**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
MYSEA: The Monterey Security Architecture	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Naval Postgraduate School,Department of Computer Science,Monterey,CA,93943**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT
see report

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**10**	

2. A remote *trusted path* mechanism that assures unambiguous user communication with the trusted computing base and is independent of client workstation security.

3. Techniques for *vertical integration of security policy* control functions with underlying security services in a Quality of Security Service framework.

4. *Secure integration of 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 resources on the multilevel server.

The remainder of this paper begins with a description of the concepts and requirements that provide the basis for the MYSEA architecture. This is followed by details regarding the structural aspects of primary MYSEA components. In Section 4, we discuss related work. We close with our conclusions and a brief description of some of the future plans for MYSEA.

2. CONCEPTS AND REQUIREMENTS

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 MYSEA network architecture affords users the ability to securely access information across networks at different classifications using standardized commodity applications. Highly trustworthy MLS servers provide the locus of security policy enforcement, while the highly trustworthy Trusted Path Extensions (TPEs) and Trusted Communications Modules (TCMs) authenticate/disambiguate users and single level networks, respectively. The TPE is a gatekeeper for user interaction with the MYSEA Server; whereas the TCM controls the access of single-level networks to the Server. Other system components provide users the ability to run unmodified office productivity tools, web-based services, and DoD applications. Federated servers support scalability, which, when combined with single sign-on and virtualization, result in a extensible computing environment.

The major components of the architecture are shown in Figure 1 [55] and include:

- High assurance MYSEA Servers, which provide the locus for multilevel security policy enforcement and host various open source or commercial application protocol servers.

- Client workstations executing popular software applications; and TPEs, which interface between the client and the MYSEA Server, providing trustworthy network security, identification and authentication, and policy support.

- Existing classified single level networks connected to the MYSEA Server via TCMs and link encryptors. TCMs complement link encryption by ensuring proper labeling of data passed back to the MYSEA Server.

- Single level servers in the Multilevel Enclave area provide application services to both local clients and those in the legacy networks.

The MYSEA Server enforces a unified mandatory access control policy for both confidentiality (including read-down) and integrity. With this basis, MYSEA provides services to support high assurance remote client authentication, session management, and connection to legacy single level networks. Users also have access to a set of application services, including SMTP, IMAP, and HTTP, which run on the MYSEA Server. Support for regrading policies is implemented in trusted applications that are constrained by the underlying reference validation mechanism. Multiple intercommunicating MYSEA Servers provide scalability within the security policy perimeter.

2.1 Usage Scenario

End users operate at client computers on the local multilevel LAN as well as on the legacy single level networks (see Coalition, SIPRNet, and NIPRNet Enclaves in Figure 1).

On the MLS LAN, the Trusted Path Extension provides a *trusted path* by which users log on to the MYSEA system. The TPE is a special purpose high assurance component inserted between the untrusted client workstation and the MLS LAN. TPE log-on establishes an identity for audit and access control purposes; then the user negotiates a session level from the range of security levels bounded by his clearance. The session level determines the domain of data and resources accessible to the user during that session, per the MLS policy.

Subsequent to session level negotiation, the user can then log on to the client workstation and use its software (e.g., web browser, e-mail client applications, or various office productivity tools) locally or to access several types of remote services: MLS services on the MYSEA Server, single-level services hosted on servers in the local multilevel enclave, and single-level services hosted on servers in the remote single level networks. To meet object reuse requirements [53], client state is purged at the end of each session, and data created or modified on the clients is stored on the MYSEA Server.

At any time, the user can invoke the trusted path to request a session level change, log off, etc. The Trusted Path Extension blocks access to the network while the user's security attributes are in flux during such operations.

Legacy network users are authenticated in their remote environment, the senstivity of which establishes their session levels for actions in the MYSEA environment. These remote users are provided two types of services: MLS services on the MYSEA Server, and single-level services hosted on servers in the multilevel enclave.

2.2 Threats and Requirements

The threats that MYSEA is designed to address fall into two major classes [34]: developmental threats and operational threats. The former includes insiders who intend to subvert the system, while the latter fall into three subclasses: network threats, malicious software, and user or application misbehavior.

Developmental threats include errors made by the development team or the malicious insertion of unintended functionality by adversaries, both of which undermine or subvert the ability of the system to protect itself from tampering or to continuously enforce critical security policy [2, 52].

Network threats are attacks to the communications protocols within the MLS LAN or the MLS Enclave. For exam-

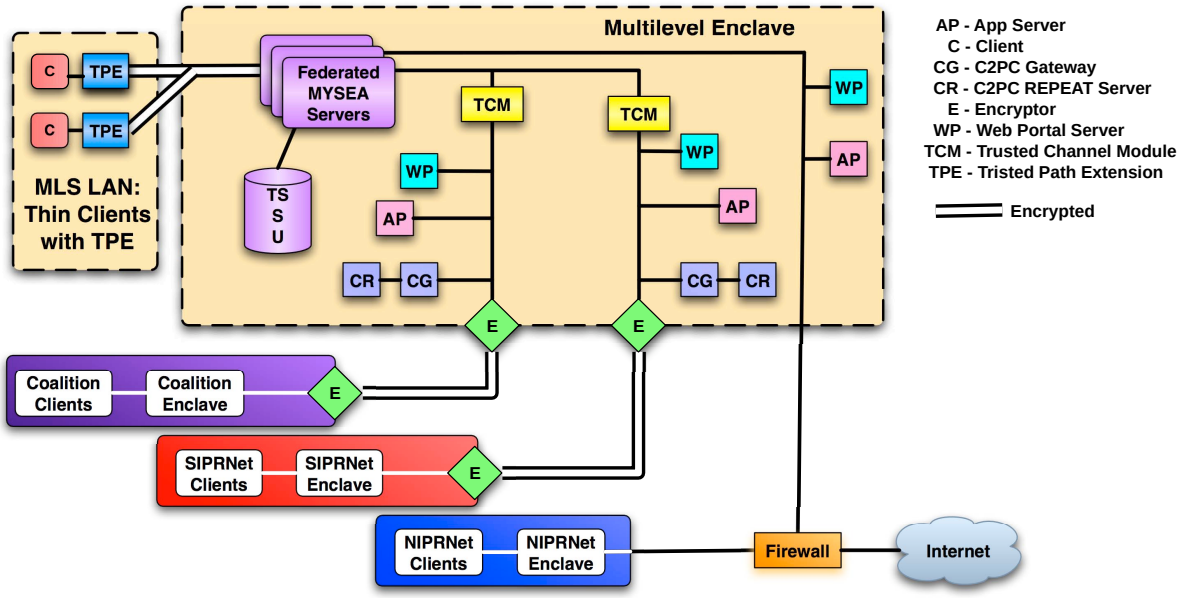

Figure 1: Monterey Security Architecture

ple, 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 may 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 [43].

Misbehavior applies to user actions at the client workstation that may initiate or participate in attacks. In this case, either users or their software attempt to bypass the TCB Extension in order to gain unauthorized access to protected information.

The system requirements are constructed to address these threats (see Appendix A in [34]). Next, we present system requirements in three categories: access control policies, assurance, and security features.

2.3 Access Control Policies

MYSEA supports both mandatory access control (MAC), and discretionary access control (DAC):

1. MAC – under this policy, the MYSEA Server enforces lattice-based policies, such as the national rules regarding access to classified information represented by the TOP SECRET, SECRET, and UNCLASSIFIED sensitivity labels. This requirement is met by the separation of resources into equivalence classes, and explicitly allowed flows between classes.

2. DAC – under this policy, MYSEA Server restricts the ability of processes, acting on behalf of users, to access data objects such as files. The access decisions are based on the MYSEA user identities associated with the processes, and permissions that are registered with the Server. Applications and users may modify permissions via runtime functions.

2.4 Assurance

Our rigorous security engineering and development processes [34] are intended to provide robust assurance of policy enforcement by the trusted components in the MYSEA system. Under this process, the threat model and system requirements specifications form the basis for the system architecture. From these, we derive functional specifications and corresponding detailed design specifications, source code, verification and testing. A key element of system assurance is the allocation of policy to components that occurs during the design process.

2.4.1 High Assurance Allocation of Policy

In the design of a distributed system architecture intended to meet high assurance requirements, the definition of both the TCB perimeter and the allocation of security policy to various components permits the system to be decomposed into analyzable components. A number of basic principles and security engineering notions come into play. Among these are the ordering of security dependencies, the ordering of trust and trustworthiness [45], and the principle of least privilege [65] as manifested in the organization of components that enforce or support the enforcement of policy, as well as components that are trusted with respect to policy.

Security dependencies in the system should be partially ordered, since if the dependencies are circular then an increasingly large component must be examined in its totality to determine if it maintains the desirable security properties. Furthermore, for their correct operation, partially ordered components will trust the components upon which they depend. If the components that are being depended upon are not trustworthy, i.e., demonstrates some measurable level of compliance with its stated functionality, then no amount of trustworthiness in the dependent component can remedy the architectural flaw created by an incorrect dependency.

If an overall security policy can be decomposed into discrete goals (e.g., rule sets); and the system design is such

that the enforcement of each individual goal can be allocated to one or more operational components, the resulting security architecture will manifest a composition of policy subsets (e.g., "TCB subsets" in [70]). This organization supports a realization of the principle of least privilege, whereby no subset implementation is endowed with more authority (or commensurate trust) than is necessary to perform its allocated function(s).

The design of architectures that reflect the notions described above is currently a non-formulaic process, and requires careful consideration over and above following established security engineering principles. One must understand how component composition may affect the ability of the system to properly enforce each of its constituent policies. For example, improper organization may permit components enforcing "weaker policies", like DAC, to deny those enforcing stronger policies, like MAC, from meeting their specified requirements.

2.5 Security Features

Major security features that MYSEA is designed to support include:

1. Secure connections to classified networks
2. Centralized security management
3. Use of adaptive security techniques to provide dynamic security services
4. True multilevel access to data at multiple levels of security using a single commodity workstation
5. Integration of multilevel security with existing sensitive networks
6. High assurance trusted communication channels to classified networks
7. Secure single sign-on across multiple MLS servers
8. Server replication to support scalability
9. IPv6 in a multilevel context
10. Interoperability with the DoD PKI infrastructure
11. High assurance trusted path techniques for managing access to classified networks

In addition to the obvious features listed above, two other features are descibed in more detail below.

2.5.1 Policy-Aware Applications

Policy-aware means that an application has been modified to run in a given policy environment without needing extraordinary privileges, and is then both fully functional and constrained by that policy [33]. Usually, applications do not need to be policy aware, and are suitably constrained by the underlying policy mechanisms. However, in some environments, such as those with MLS policies, application modifications are required.

MYSEA's HTTP server, which includes an MLS-enabled wiki, is a concrete example of a policy-aware application. In the wiki, multilevel technology permits collaborators with different security attributes in a coalition environment to maximize information sharing while still adhering to the constraints of the overall security policy [58]. Multilevel-aware instances of the wiki execute as untrusted subjects within the context of a multilevel architecture. MLS policy is enforced with high assurance by the MYSEA Server, which ensures that wiki users logged in at high sensitivity levels are able to read and post information at their level, but may not write information to lower sensitivity levels. Correspondingly, users at lower sensitivity levels are only allowed access to less sensitive information.

2.5.2 Dynamic Security Services

Complex and adaptive networks may require changes on demand to the security provided. When conditions on the network change, requirements for security — e.g., restrictions as seen from the users' or attackers' point of view — may also change. In MYSEA, the DSS Quality of Security Service (QoSS) mechanisms located on the TPE and at the MYSEA Server can modify the protection services afforded to an ongoing session, in response to a change notification. 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 [13] permits selection of protection mechanisms.

3. MYSEA DISTRIBUTED STRUCTURE

The MYSEA Server implements both multilevel and discretionary security policies while maintaining support for new and legacy applications and unmodified commodity client systems. The architecture supports protocols and equipment from a wide range of vendors as well as secure interaction with external classified networks. The mandatory access control (MAC) security policy is based on the Bell and LaPadula [10] confidentiality policy and the Biba [12] integrity policy.

The access-control policy foundation for the MYSEA Server is the BAE XTS-400 [5], which has been awarded a Common Criteria [18] EAL5 certification (see http://www.niapccevs.org/cc-scheme/st/vid10293) for its combined hardware base and STOP operating system. MYSEA extends the XTS-400 with an MLS network interface and TCP/IP stack, remote trusted path, remote file system, and remote interactive shell capability.

The MYSEA software architecture is illustrated in Figure 2. The MYSEA Server, TPE and TCM components all have a common foundation: a high assurance operating environment (i.e., LPSK [46] and STOP OS), the Protected Communications Service (PCS) and DSS.

The PCS component provides IPsec-based protected communication channels between the TPE and server, and between the TCM and the server. The DSS components implement a dynamic service management mechanism that can adjust PCS protection in response to external changes and threats, as envisioned for the Global Information Grid (GIG) [79].

The Trusted Path Service (TPS) and Trusted Path Application (TPA) components together enforce the identification and authentication supporting policy to ensure that only authorized users can gain access to the system. The TPA affords the users unspoofable access to security critical services and is invoked via a Secure Attention Key. The TPS com-

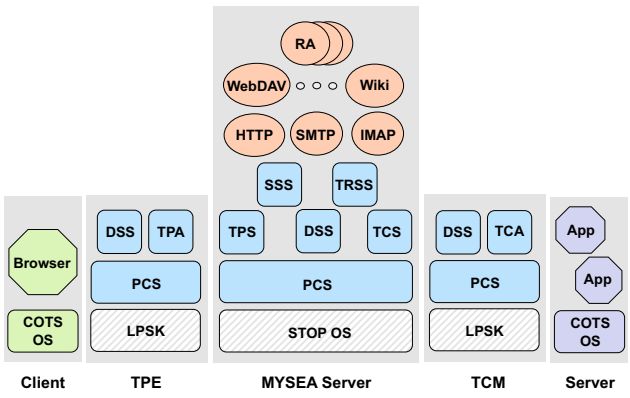

Figure 2: MYSEA Software Architecture

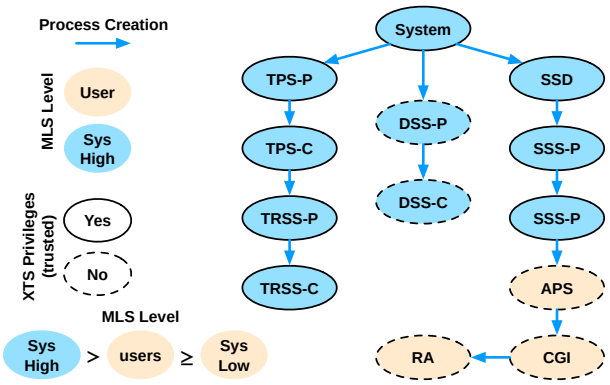

Figure 3: MYSEA Server Processes

ponent on the server handles user authentication and session negotiation functions. Likewise, the Trusted Channel Service (TCS) and the Trusted Channel Application (TCA) components ensure that traffic between a single level network and the MYSEA Server are properly labeled at the classification level of the particular network.

3.1 Trusted Path Extension and Trusted Channel Module

The Trusted Path Extension (TPE) and Trusted Communications Module (TCM) help to enforce policies under the direction of the MYSEA Server, but neither is empowered to make policy decisions.

When a user logs in, the TPE passes the user's identity credentials to the MYSEA Server, which validates the login attempt and instructs the TPE whether to allow or deny access to the network. In negotiating a session level, the TPE passes the user's session level request to the Server for a decision. After a successful login and session level negotiation, the TPE allows the user to access the MLS LAN and the MYSEA Server as well as services in the Multilevel Enclave and the single level networks.

Similarly, the Trusted Channel Module (TCM) helps to map the single level networks to specific security levels, ensuring for example, that "IP-spoofing" could not be used by a malicious remote user to access the wrong level of information. Users on these networks can only access data on the MYSEA Server at the classification level of the single level network from which they are operating.

3.2 Workstations

The use of a single client workstation for cross-domain access provides a dramatic physical footprint reduction compared to other approaches. However, without appropriate security measures, use of a single workstation could magnify the risk of information leakage. In particular, residual information in memory or other internal components of the workstation allocated during a high session may be improperly reallocated to a low session. To address these *object reuse* issues, MYSEA uses stateless, i.e. diskless, clients. All user data objects and related metadata are stored on the MLS server rather than on the workstation. To avoid object reuse with respect to state elements on the client workstation, users recycle workstation power whenever they transition to a less sensitive session. *Boot odometer* support [75] could monitor this procedure.

3.3 DSS Tool

The DSS tool manages the IPsec configurations available to the MYSEA components. At initialization, the DSS Admin Tool connects to the DSS Server and thereafter, it conveys users' input to the DSS Server. Its functions include: Set Policy; Reload Policies; get and display Server IKE Security Association Database information; and get and display Server IPSec Security Policy Database information.

3.4 MYSEA Server

The MYSEA Server comprises a set of interacting processes that implement various multilevel services and single-level applications. As shown in Figure 3, trusted, multilevel processes control the invocation of single-level applications that interact with the user at the user's current session level.

3.4.1 Multilevel Services

MYSEA offers the following multilevel services: Trusted Path, Secure Session, Trusted Remote Session, and Dynamic Security. Single-level application support is provided by Application Protocol Servers and Remote Applications, along with support from CGI processes.

Trusted Path Service

The *Trusted Path Service* is provided by a single TPS parent and multiple TPS child processes. The parent process is started at initialization and monitors the network for TPE connection requests. If a request comes from a valid TPE, the TPS parent creates a TPS child process to service the connection (e.g., handle TPE requests to login, change session level, run, and logout). The TPS parent and child processes execute at the system network security level (viz., system high) and have privileges to access system identification and authentication information, and both MAC and DAC bypass privileges.

The TPS child processes also create the Trusted Remote Session Server (TRSS) parent process for each user name and session level pair (see below); and support the DSS server and DSS Clients with respect to various user/TPE activities. The TPS child alerts the DSS Server when it encounters an error and when the user has initiated a RUN or Logout command. Depending on the situation, the DSS server may take various actions, such as changing the security configuration of the DSS Client, or blocking the TPE.

Secure Session Service

At start-up, the *Secure Session Daemon (SSD)* process starts a *Secure Session Server (SSS)* parent process for each supported application protocol. Each SSS parent accepts and validates application protocol service requests from TPE clients for its particular TCP/IP protocol (IMAP, SMTP, HTTP, etc). The parent process ensures that the request is from either a TPE with a valid session, or a valid Remote Application, and it creates an SSS child process to service the connection. After creating the child, the parent continues to monitor for new application protocol service requests.

The SSS child process creates an *Application Protocol Server (APS)* process and handles communications between the client TPE (or Remote Applications) and the APS. The APS process is created with the privileges of the user session responsible for the TPE or RA request. The SSS child and the APS processes communicate via a MYSEA Socket which is allocated by the TPS child process.

Trusted Remote Session Service

The *Trusted Remote Session Service (TRSS)* processes handle communication requests from Remote Applications. There is a TRSS parent process for each user name and session level pair — e.g., a given user might have sessions at the same level on different client workstations, all of which would be serviced by the same TRSS parent. The TRSS parent process creates a TRSS child for each new remote connection request from a Remote Application (RA). The TRSS child process handles all remote connections with the RA.

For each connection established for the RA to the destination, the TRSS child process updates a database to associate a user ID and session level with a particular destination IP, destination port, source IP and source port.

The TRSS processes execute at the system network security level, and they have privileges to communicate with remote applications, and to be started by a trusted process.

Dynamic Security Service

The *Dynamic Security Service (DSS)* include a DSS Server Parent process, one or more DSS Server child processes, one or more DSS Clients, and the DSS tool (see 3.3 for a description of the latter). The DSS Clients and DSS Tool are typically on different nodes of the network than the MYSEA Server.

DSS Server parent handles communication requests from DSS Clients. It creates a TRSS child for each new connection request, ensuring that there is only one connection for each DSS Client. The DSS Server parent also handles commands from the DSS Admin Tool and the TPS process — e.g., to change the dynamic parameter or load a new configuration — to which it responds by passing these commands to the appropriate DSS Server child.

The DSS Clients coordinate security policies with the PCS components, manage IKE daemons, and take direction from the DSS child processes. The DSS child may request: the load/unload of a security configuration, restart of its IKE daemon, and the return of IKE/IPSec information.

3.4.2 Application Invocations

Applications may be invoked on the MYSEA Server from client workstations, as well as from components on the server.

The *Application Protocol Server (APS)* process implements the server side of an application level client/server protocol. MYSEA provides the following APS services: HTTP (Apache), WebDav, IMAP, SMTP, and MLS Wiki. An APS process communicates with the client via a MYSEA socket managed by an SSS child process. The program for an APS process is an implementation of an industry standard application protocol, which is *policy aware*, i.e., it has been modified to allow it to interact in a multilevel environment and to use the MYSEA socket infrastructure; e.g., calls to *socket* in the APS are changed to *mskt_socket*.

One particular APS, using the HTTP protocol, supports a menu of *Remote Applications* from which the user at the client workstation can choose. In response to such a choice, the HTTP APS invokes a simple *CGI process*, which initiates the chosen Remote Application.

Users can launch certain interactive shell sessions via the WebShell CGI program, and can use the MYSEA WebDAV APS to navigate the MYSEA Server's file structure (e.g., home directories and the Apache document root).

The Remote Application (RA) is an application program executing on the server on behalf of the client. As with the APS processes, the program for an RA process is an implementation of an industry standard application protocol, which has been modified to allow it to interact in a multilevel environment and to use the MYSEA socket infrastructure. An RA process communicates with the client via a MYSEA Socket managed by a TRSS child process. For example, when a RA wants to establish a new remote connection, it signals the TRSS parent process, which starts a TRSS child process to handle the connection.

MYSEA includes the Trivial File Transfer Protocol (TFTP) remote application.

4. RELATED WORK

Hinke suggested the idea of a high assurance server to provide a locus of multilevel secure control to single level clients [31]. In his 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 [64] 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 [14, 42, 7] 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 [29], 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 [42], it was never fielded. The VMM approach remains problematic for separation of different domains of data because of the difficulty of creating a trusted VMM. This task is made even more difficult because many current microprocessors are not *strictly* virtualizable [62], which increases the complexity of software.

Non-distributed approaches to support access to multilevel data via COTS applications have been proposed in Seaview and some VMM architectures [22, 50, 60]. 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 [27] 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 the number of documents to be replicated and the number of security levels to which documents are replicated, they are untenable. 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 avoids these problems.

The Naval Research Laboratory (NRL) Network Pump [40] allows messages from a low sensitivity level to be sent to a high sensitivity level, and prohibits 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 more general two-way connectivity between multiple subnets at different sensitivity levels, resulting in a multiple single-level (MSL) network [41]. The capital and administrative cost of separately maintained LANs is a drawback that the MYSEA avoids.

Starlight [3] 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 the 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.

4.1 Other Multilevel Variations

The ruleset based access control (RSBAC) system [59] 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 both the calling subject and the target to be accessed. MYSEA's DSS mechanism allows both a more fine-grained and a more dynamic security policy.

The Security-Enhanced (SE) Linux project is an approach to controlling multiple information domains in an open source operating system [49, 71]. 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 [66] relies on various authentication and cryptographic technologies to mediate user's access to information, but like the other variants discussed in this section provides no underlying basis of trust to ensure against subversion or malicious software that might corrupt or leak information.

4.2 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 [51], Trusted Solaris [72], and XTS-400 [76] 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 itself does not support a remote trusted path.

4.3 Dynamic Security Services

Dynamic Security Services (sometimes referred to as "Quality of Security Service," [35]) refers to offering variable levels of security to both users and tasks in support of increasing system *quality*. Thus, security is transformed from a performance obstacle into an adaptive, constructive network management parameter.

Historically, there have been several efforts in this direction. A quality of protection parameter is provided in the GSSAPI specification [48]. This parameter is intended to manage the level of protection provided to a message communication stream by an underlying security mechanism (or service), "allowing callers to trade off security processing overhead dynamically against the protection requirements for particular messages." Another early reference to a variable security service is that of Schneck and Schwan [67], which discusses variable packet authentication rates with respect to the management of system performance. References to security in the QoS literature can be found in [19, 4, 77], although little is mentioned there of security as a functional QoS dimension.

5. CONCLUSIONS AND FUTURE WORK

The need for high-assurance architectures that implement multi-domain information protection mechanisms is widespread and growing. However, such architectures will not be adopted unless they provide users with currently required functionality, the ability to easily incorporate new applications and software updates, and a familiar interface.

MYSEA is a trusted distributed operating environment for enforcing multi-domain security policies that supports

unmodified COTS productivity applications in support of usability. The architecture encompasses a combination of many (untrusted) commercial components and relatively few trusted multi-domain components. MYSEA introduces several innovations for protecting multilevel data and for managing security policies and security services in support of critical applications, including:

1. A distributed high assurance architecture for controlling access to multiple data domains, which utilizes commercial and open source applications
2. A high assurance distributed trusted path mechanism
3. Access to existing single-level networks
4. A QoSS framework providing dynamic (adaptive) security services

It is hoped that the development of high-assurance, highly usable MLS architectures such as MYSEA will encourage the adoption of MLS computing systems by the entities that stand to benefit from their use.

In the future, we intend to support new applications including voice mail, video telephones, and webmail. We are also exploring multilevel aware collaboration services, and the addition of QoSS to services other than network security. Additional future work includes a Network File System (NFS) port enhanced with ring-like privileges [68] in the user domain, to help constrain the behavior of applications, and a multilevel aware or multilevel DNS service [20].

6. CONTRIBUTIONS AND ACKNOWLEDGMENTS

This paper is an overarching description and summary-to-date of MYSEA. As a team effort, it is the collective contributions of a cohesive group to which each member brings individual talent that have resulted in our ongoing progress. Irvine is lead scientist for the MYSEA project; she invented its structural foundation and continues to guide its design and development. Nguyen supervises MYSEA engineering, is a key member of the design team, and led the effort to document and provide configuration management for the design. Shifflett has participated in the MYSEA project in the areas of subsystem design and implementation. Levin has contributed analysis and advice, particularly at the level of conceptual design elements. Khosalim and Prince have made ongoing contributions to the evolution of MYSEA, especially in of implementation and testing. Clark made early contributions to the project. Gondree recently joined the MYSEA team in the areas of design and architecture.

A continuum of talented graduate students have contributed to MYSEA under the supervision of Irvine, with the support of MYSEA team members as co-advisors and second readers [6, 8, 9, 11, 15, 16, 17, 21, 23, 24, 25, 26, 28, 30, 44, 56, 57, 61, 63, 69, 73, 74, 78]

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

7. REFERENCES

[1] AFCEA. The Need to Share: The U.S. Intelligence Community and Law Enforcement. http://www.afcea.org/mission/intel/documents/SpringIntel07whitepaper_000.pdf (Last checked: 3 Aug 2009), Fairfax, VA, April 2007.

[2] 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).

[3] M. Anderson, C. North, J. Griffin, R. Milner, J. Yesberg, and K. Yiu. Starlight: Interactive link. In *Proc. 12th Computer Security Applications Conf.*, San Diego, CA, December 1996.

[4] C. Aurrecoechea, A. Campbell, and L. Hauw. A Survey of Quality of Service Architectures. *Multimedia Systems Journal*, 1996.

[5] L. BAE Systems Information Technology. *Security Target, Version 1.11 for XTS-400 Version 6.* BAE, December 2004.

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

[7] S. R. Balmer and C. E. Irvine. Analysis of Terminal Server Architectures for Thin Clients in a High Assurance Network. In *Proc. National Information Systems Security Conf.*, pages 192–202, Baltimore, MD, October 2000.

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

[9] T. J. Baumgartner and M. D. W. Phillips. Implementation of a Network Address Translation Mechanism Over IPv6. Master's thesis, Naval Postgraduate School, Monterey, CA, June 2004.

[10] D. Bell and L. La Padula. Secure computer systems: A mathematical model. Technical Report MTR-2547, Vol 2, MITRE Corp., Bedford, MA, Nov. 1973.

[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] K. Biba. Integrity considerations for secure computer systems. Technical Report TR-3153, Mitre, Bedford, MA, Apr. 1977.

[13] M. Blaze, J. Feigenbaum, and A. D. Keromytis. KeyNote: Trust Management for Public-Key Infrastructures. In *Proc. 1998 Security Protocols International Workshop*, pages 59–63, Cambridge, England, April 1998. Springer LNCS vol. 1550.

[14] T. Borden, J. Hennessy, and J. Rymarczyk. Multiple Operating Systems On One Processor Complex. *IBM Systems Journal*, 28(1):104–123, 1989.

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

[16] 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.

[17] S. Bui. Single Sign-On Solution For MYSEA Services. Master's thesis, Naval Postgraduate School, Monterey, California, September 2005.

[18] CCMB. *Common Criteria for Information Technology Security Evaluation*. Number CCMB-2006-09-001. Common Criteria Maintenance Board, 3.1 revision 1 edition, September 2006.

[19] S. Chatterjee, B. Sabata, and J. Sydir. ERDoS QOS Architecture. Technical Report ITAD-1667-TR-98-075, SRI Intl., Menlo Park, CA, May 1998.

[20] P. C. Clark, T. E. Levin, C. E. Irvine, and D. J. Shifflett. DNS and Multilevel Secure Networks. Technical report, Naval Postgraduate School, Monterey, California, February 2009.

[21] R. C. Cooper. Remote Application Support in a Multi-Level Environment. Master's thesis, Naval Postgraduate School, March 2005.

[22] 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 *Proc. 1988 IEEE Symposium on Security and Privacy*, Oakland, CA, April 1988.

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

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

[25] M. Egan. An Implementation Of Remote Application Support In A Multilevel Environment. Master's thesis, Naval Postgraduate School, Monterey, California, March 2006.

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

[27] J. Froscher, M. Kang, J. Mcdermott, O. Costich, and C. E. Landwehr. A Practical Approach to High Assurance Multilevel Secure Computing Service. In *Proc. Computer Security Applications Conf.*, pages 2–11, Orlando, FL, December 1994.

[28] C. Gilkey. Proof of concept integration of a single-level service-oriented architecture into a multi-domain secure environment. Master's thesis, Naval Postgraduate School, Monterey, CA, March 2008.

[29] R. Goldberg. *Architectural Principles for Virtual Computer Systems*. PhD thesis, Harvard University, Cambridge, MA, 1972.

[30] 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.

[31] T. Hinke. The Trusted Approach to Multilevel Security. In *Proc. Computer Security Applications Conf.*, pages 335–341, December 1990.

[32] IRTPA. Intelligence reform and terrorism prevention act of 2004. http://thomas.loc.gov/cgi-bin/query/D?c108:4:./temp/ c108PvI049::, 28 January 2004.

[33] C. E. Irvine, T. Acheson, and M. F. Thompson. Building Trust into a Multilevel File System. In *Proc. 13th National Computer Security Conf*, pages 450–459, Washington, DC, October 1990.

[34] 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*, 7(4):192–208, 2002.

[35] C. E. Irvine and T. E. Levin. Quality of Security Service. In *Proc. New Security Paradigms Workshop*, pages 91–99, Balleycotten, Ireland, September 2000.

[36] C. E. Irvine, T. E. Levin, T. D. Nguyen, D. Shifflett, J. Khosalim, P. C. Clark, A. Wong, F. Afinidad, D. Bibighaus, and J. Sears. Overview of a High Assurance Architecture for Distributed Multilevel Security. In *Proc. 2004 IEEE Systems Man and Cybernetics Information Assurance Workshop*, pages 38–45, West Point, NY, June 2004.

[37] C. E. Irvine, D. J. Shifflett, P. C. Clark, T. E. Levin, and G. W. Dinolt. MYSEA Security Architecture. Technical Report NPS-CS-02-006, Naval Postgraduate School, Monterey, CA, May 2002.

[38] C. E. Irvine, D. J. Shifflett, P. C. Clark, T. E. Levin, and G. W. Dinolt. Monterey Security Enhanced Architecture Project. In *DARPA DISCEX Conf.*, pages 176–181, April 2003.

[39] C. E. Irvine, D. J. Shifflett, P. C. Clark, T. E. Levin, and G. W. Dinolt. MYSEA Technology Demonstration. In *DARPA DISCEX Conf.*, volume II, pages 10–12, April 2003.

[40] M. H. Kang, J. N. Froscher, and B. J. Eppinger. Towards and Infrastructure for MLS Distributed Computing. In *Proc. 14th Annual Computer Security Applications Conf.*, pages 91–100, Phoenix, AZ, December 1998.

[41] M. H. Kang and I. Moskowitz. Design and Assurance Strategy for the NRL Pump. *IEEE Computer*, 31(4):56–64, April 1998.

[42] 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 *Proc. IEEE Symposium on Research on Security and Privacy*, pages 2–19, Oakland, CA, 1990.

[43] B. Lampson. A Note on the Confinement Problem. *CACM*, 16(10):613–615, 1973.

[44] C. Lavelle. A preliminary analysis for porting XML-based chat to MYSEA. Master's thesis, Naval Postgraduate School, Monterey, California, June 2008.

[45] T. E. Levin, C. E. Irvine, T. V. Benzel, G. Bhaskara, P. C. Clark, and T. D. Nguyen. Design Principles and Guidelines for Security. Technical Report NPS-CS-07-014, Naval Postgraduate School, Monterey, California, November 2007.

[46] T. E. Levin, C. E. Irvine, and T. D. Nguyen. Least privilege in separation kernels. In J. Filipe and M. S. Obaidat, editors, *E-business and Telecommunication Networks; Third International Conference, ICETE 2006, Set'ubal, Portugal, August 7-10, 2006,*, volume 9 of *Communications in Computer and Information Science*. Springer, 2008.

[47] T. E. Levin, C. E. Irvine, and E. Spyropoulou. *Quality of Security Service: Adaptive Security*, volume 3, pages 1016–1025. John Wiley and Sons, Hoboken, NJ, January 2006.

[48] J. Linn. Generic Security Service Application Program Interface Version 2, Update 1, 2000.

[49] P. Loscocco and S. Smalley. Integrating Flexible Support for Security Policies into the Linux Operating System. Technical report, National Security Agency, October 2001.

[50] T. F. Lunt, R. R. Schell, W. Shockley, M. Heckman, and D. Warren. A Near-Term Design for the SeaView Multilevel Database System. In *Proc. IEEE Symposium on Security and Privacy*, pages 234–244, Oakland, 1988.

[51] Microsoft. Windows 2000 Evaluated Configuration Administrator's Guide, Version 1.0. Technical report, Microsoft Corporation, Redmond, WA, 2002.

[52] P. Myers. Subversion: The Neglected Aspect of Computer Security. Master's thesis, Naval Postgraduate School, Monterey, CA, 1980.

[53] NCSC. A guide to understanding object reuse in trusted systems. Technical Report NCSC TG-018, National Computer Security Center (NCSC), Fort George G. Meade, MD, 1991.

[54] T. D. Nguyen, C. E. Irvine, and T. E. Levin. A Testbed for High Assurance and Dynamic Security. Technical Report NPS-CS-08-010, Naval Postgraduate School, Monterey, CA, May 2008.

[55] T. D. Nguyen, T. E. Levin, and C. E. Irvine. MYSEA testbed. In *Proc. 6th IEEE Systems, Man and Cybernetics Information Assurance Workshop*, pages 438–439, West Point, NY, June 2005.

[56] M. O'Neal. A Design Comparison Between IPv4 and IPv6 in the Context of MYSEA, and Implementation of an IPv6 MYSEA Prototype. Master's thesis, Naval Postgraduate School, Monterey, CA, June 2003.

[57] K. L. Ong. Design and Implementation of Wiki Services in a Multilevel Secure Environment. Master's thesis, Naval Postgraduate School, Monterey, California, December 2007.

[58] K. L. Ong, T. D. Nguyen, and C. E. Irvine. Implementation of a Multilevel Wiki for Cross-Domain Collaboration. In *Proc. Third International Conf. on i-Warfare and Security*, pages 293–304, Omaha, NB, April 2008.

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

[60] B. Pomeroy and S. Weisman. Private Desktops and Shared Store. In *Proc. 14th Computer Security Applications Conf.*, pages 190–200, Phoenix, AZ, December 1998.

[61] A. D. Portner. A prototype of multilevel data integration in the MYSEA testbed. Master's thesis, Naval Postgraduate School, Monterey, California, September 2007.

[62] J. S. Robin and C. E. Irvine. Analyzing the Intel Pentium's Capability to Support a Secure Virtual Machine Monitor. In *Proc. 9th USENIX Security Symposium*, Denver, CO, August 2000.

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

[64] J. Rushby and B. Randell. A Distributed Secure System. In *Computer*, pages 55–67, May 1983.

[65] J. H. Saltzer and M. D. Schroeder. The protection of information in computer systems. *Proc. IEEE*, 63(9):1278–1308, 1975.

[66] C. Sanders. Information Support to Multinational Operations. *The Edge*, 5(2), July 2001.

[67] P. A. Schneck and K. Schwann. Dynamic Authentication for High-Performance Networked Applications. Technical Report GIT-CC-98-08, Georgia Institute of Technology College of Computing, 1998.

[68] M. D. Schroeder and J. H. Saltzer. A hardware architecture for implementing protection rings. *Comm. A.C.M.*, 15(3):157–170, 1972.

[69] J. D. Sears. Simultaneous Connection Management and Protection in a Distributed Multilevel Security Environment,. Master's thesis, Naval Postgraduate School, Monterey, CA, September 2004.

[70] W. R. Shockley and R. R. Schell. TCB subsets for incremental evaluation. In *Proc. Third AIAA Conf. on Computer Security*, pages 131–139, December 1987.

[71] S. Smalley and T. Fraser. A Security Policy Configuration for Security-Enhanced Linux. Technical report, NAI Labs, January 2001.

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

[73] T. F. Tenhunen. Implementing an Intrusion Detection System in the MYSEA Architecture. Master's thesis, Naval Postgraduate School, Monterey, California, June 2008.

[74] R. C. Vernon. A design for sensing the boot type of a trusted platform module enabled computer. Master's thesis, Naval Postgraduate School, Monterey, California, September 2005.

[75] R. C. Vernon, C. E. Irvine, and T. E. Levin. Toward a boot odometer. In *Proceedings from the 7th IEEE Systems, Man and Cybernetics Information Assurance Workshop*, West Point, NY, June 2006.

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

[77] L. Welch, M. W. Masters, L. Madden, D. Marlow, P. Irey, P. Werme, and B. Shirazi. A Distributed System Reference Architecture for Adaptive QoS and Resource Management. In J. Rolim, editor, *Proc. 11th IPPS/SPDP'99 Workshops*, pages 1316–1326, Berlin, April 1999. Springer.

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

[79] P. Wolfowitz, "Global Information Grid (GIG) over arching policy." U.S. Department of Defense, directive number 8100.1, 19 September 2002.