A Cloud-Oriented Cross-Domain Security Architecture

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

***Abstract*—The Monterey Security Architecture addresses the need to share high-value data across multiple domains of different classification levels while enforcing information flow policies. The architecture allows users with different security authorizations to securely collaborate and exchange information using commodity computers and familiar commercial client software that generally lack the prerequisite assurance and functional security protections. MYSEA seeks to meet two compelling requirements, often assumed to be at odds: enforcing critical, mandatory security policies, and allowing access and collaboration in a familiar work environment.**

Recent additions to the MYSEA design expand the architecture to support a cloud of cross-domain services, hosted within a federation of multilevel secure (MLS) MYSEA servers. The MYSEA cloud supports single-sign on, service replication, and network-layer quality of security service. This new cross-domain, distributed architecture follows the consumption and delivery model for cloud services, while maintaining the federated control model necessary to support and protect cross-domain collaboration within the enterprise. The resulting architecture shows the feasibility of high-assurance, cross-domain services hosted within a community cloud suitable for inter-agency, or joint, collaboration. This paper summarizes the MYSEA architecture and discusses MYSEA's approach to provide an MLS-constrained cloud computing environment.

Keywords: cloud computing, cross-domain services, collaborative applications, quality of security services.

I. INTRODUCTION

In recent years, a system that provides the ability to access or transfer information between two or more security domains is referred to as a cross domain solution (CDS) [1]. In the CDS context, three technology categories have been defined by the Unified Cross Domain Management Office: *access*, *transfer*, and *multilevel* [2]. A cross domain access system allows a user to access information in different domains from a single machine but no information is transferred between domains. A cross domain transfer system controls information moving between domains, e.g., a data guard. A cross domain multilevel system manages information of different security levels stored in a common repository and enforces a mandatory security policy to control both information access and information flow. Systems in the last category are also known as multilevel secure (MLS) systems.

Access to information in an MLS system is governed by the classification level of the information, the security clearance of the requester and whether the requester has a need to access the information, i.e., the "need-to-know" caveat. Mandatory access control distinguishes an MLS system from typical secure systems, where the latter are constructed using commercial operating systems that control access to information based on security attributes that can be changed by users, e.g., access control lists and file permissions in Windows or Linux.

In addition to the MLS policy enforcement mechanism, support for robust user authentication, MLS-constrained services and dynamic security services are other desirable functionalities of a distributed MLS system architecture. However, experience has shown that applying security without considering usability leads to failure due to lack of user acceptability [3]. User-driven requirements, such as ease of learning and efficiency of use, gave impetus to our work on the Monterey Security Architecture (MYSEA) [4].

Although originally designed to address the inefficient exchange of information in military "silo" environments, MYSEA has evolved to provide new capabilities for composing secure, distributed cross-domain services and transparent access to disparate single-level networks. MYSEA's properties and capabilities have naturally developed to support an "MLS Cloud" – where features of cloud computing have been integrated with the high-assurance and strong policy enforcement required by MLS systems. Specifically, the MYSEA architecture has been designed to support agile and adaptive security provisioning, service replication, stateless "thin clients" whose data and applications are primarily provided as remote-hosted services, the sharing of costly (high-assurance) resources, and the ability to scale to support many simultaneous users. Recent extensions to the MYSEA design include security enhancements to support the vertical integration of modern collaborative applications and network-layer quality of security service that allows the federation to dynamically adapt its security posture in response to network conditions (e.g., INFOCON threat levels) and policy constraints on the user session.

The remainder of this paper describes the MYSEA system architecture, the composition of trusted and untrusted components, and a conceptual model for dynamic management of security services. The paper concludes by reviewing related

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 either the Office of Naval Research or the National Reconnaissance Office.

work and summarizing our approach to designing a cloud-oriented cross-domain computing environment.

II. Key Concepts and Technologies

The provenance of multilevel security can be partially traced back to Anderson's seminal report which introduced the concept of a reference monitor and its implementation, i.e., the reference validation mechanism, and discussed the security threats and engineering problems related to multi-user resource-sharing systems [5, 6, 7]. The notion of a security kernel implementing the reference validation mechanism quickly became a well-studied design option for secure operating systems.

Classic security kernels, e.g., PDP-11/45 [8], SCOMP [9, 10] and GEMSOS [11], were developed for high assurance trusted operating systems, and, like the Boeing MLS LAN [12] and XTS-300 [13], were certified under the now-obsolete Trusted Computer System Evaluation Criteria (TCSEC) [14]. The XTS-300 was originally certified for the second highest level of assurance defined by the TCSEC, i.e., Class B3. Its successor, the XTS-400 [15], has been certified at Evaluation Assurance Level 5 (EAL5) under the Common Criteria Version 2.3 [16]. The XTS-400's operating system, the Secure Trusted Operating Program (STOP), enforces a unified mandatory access control security policy based on the Bell and LaPadula confidentiality policy [17] and the Biba integrity policy [18]. The STOP security kernel provides the trusted foundation of the MYSEA server.

A recent trend in trusted system development has been the use of separation kernels in avionics systems [19, 20], real-time embedded systems [21] and virtual machine monitors (VMM) [22]. With respect to security evaluation, the separation kernel approach was initially discussed in the context of a separation VMM [23] and was recently defined in the Separation Kernel Protection Profile (SKPP) [24]. In addition to the basic separation kernel, the SKPP includes security requirements for a Least Privilege Separation Kernel (LPSK) [25], which supports finer-grained privilege controls based on the Principle of Least Privilege [26].

User identification and authentication (I&A) plays an important role in making access control decisions. In an information system, users must be unambiguously identified and authenticated prior to gaining access to information [27]. A secure system must afford users the ability to communicate with the trusted computing base (TCB) via a high integrity communication channel (i.e., a *trusted path*). A user can invoke the trusted path via an unspoofable *Secure Attention Key* (e.g., a special keystroke sequence) to perform security functions that require user actions (e.g., user's login, changing a user's session level) [28]. Similarly, security-relevant communications between two trusted components in a distributed system must also be protected from unauthorized modification and disclosure via a *trusted channel*. For non-repudiation purposes, the identity of the end points of a trusted channel must be uniquely authenticated [28].

In MYSEA, the TPE and TCM components support the establishment of *remote* trusted paths between users having no access to the system console (in the MYSEA cloud) and the MYSEA server, and trusted channels between the single-level network and the MYSEA server, respectively. The XTS-400 only supports *local* trusted paths between users having access to the system console and the XTS-400's TCB. The trusted foundation of the TPE and TCM components is the Trusted Computing Exemplar LPSK [29]. The TCX LPSK, designed to conform to the SKPP, manages all resources on the platform, assigns a set of exported resources to different *partitions* and controls information flows between partitions as defined by the LPSK configuration data [30].

The Internet Protocol security (IPsec) base architecture [31] and its associated protocols [32, 33, 34] have been widely used to implement cryptographic security services at the network layer. The IPsec protocols support different sets of cryptographic algorithms, modes of operations (e.g., transport, tunnel) and key management schemes (e.g., Internet Key Exchange (IKE)). IPsec is used in MYSEA to implement the protected tunnels between the trusted appliances (viz., TPEs and TCMs) and the MYSEA server.

The quality of service (QoS) of a distributed system is commonly associated with a set of parameters representing different characteristics of individual applications or of the overall system [35, 36], not as a dimension of control. Traditional QoS mechanisms afford users the ability to select different classes of service related to various functional dimensions of a system such as accessibility, reliability and performance. Security was considered as a QoS factor in a Quality of Security Service (QoSS) mechanism that is based on the concept of *variant security* [37]. This QoSS model stipulates that the QoS mechanism must be capable of modulating strength of service according to specific levels of security requested by users, a resource management system or level of threat reported by an intrusion detection system. The previous QoSS work also presents an approach for regulating IPsec cryptographic attributes (e.g., cryptographic algorithm, key length) in response to QoSS parameters such as the operating mode and security level of a network. The Dynamic Security Services (DSS) mechanism in MYSEA extends this approach to support dynamic modulation of application services (hosted on the MYSEA Servers) based on the security level of a user's session.

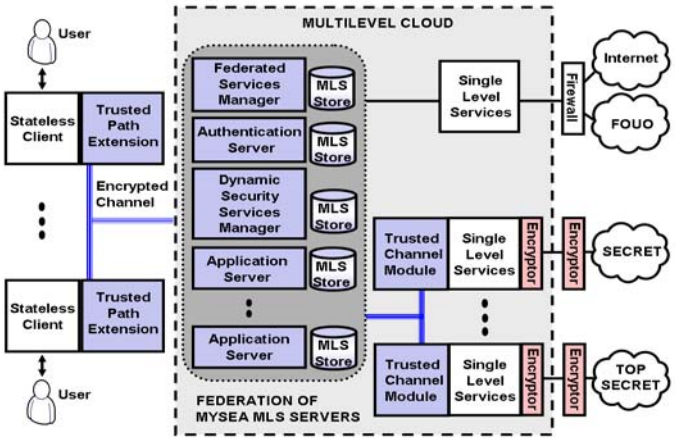

Figure 1. Monterey Security Architecture

III. Overall Architecture

MYSEA is a high assurance MLS-constrained cloud computing environment that allows authenticated users executing popular commercial applications to securely access data and services at different classification levels in the context of a single session.

The MYSEA cloud is composed of a group of collaborative MYSEA Servers, each running on a highly trustworthy commercial MLS server platform, and a small number of special purpose trusted network interface components, i.e., the Trusted Path Extension (TPE) and the Trusted Channel Module (TCM). The MYSEA Servers communicate with each other via a dedicated network and jointly provide centralized multilevel security policy enforcement. The TPE ensures unspoofable authentication of users on one or more MLS local area networks and the TCM provides high assurance labeling of incoming traffic from multiple single-level networks.

The commodity client workstations on the MLS LAN and system components on the single-level network provide users with the ability to run unmodified office productivity tools and collaborative applications. High assurance network encryptors provide protected communication channels required for high-value/classified data transmission between the MYSEA controlled environment and external networks (see Fig. 1).

A. *MYSEA Cloud Servers*

The nerve center of a MYSEA system is a cluster of MLS servers that cooperatively enforces a system-wide multilevel security policy and hosts policy-constrained application protocol servers. The MYSEA cloud relies on four fundamental components.

The *Federated Services Manager* handles queries about user sessions (required to implement single sign-on) and service availability if such information is not available on the local MYSEA Server. The *Authentication Server* enforces the identification and authentication supporting policy to ensure that only authorized users can gain access to the system. The *Dynamic Security Services Manager* implements a service management mechanism that can adjust to changing operational needs and situational threats. The *Application Server* handles application requests from the MYSEA clients. Supported services include web browsing, wiki, WebDAV, email, webmail, VoIP direct call and voice mail. Each service can be hosted on the same or separate MYSEA Servers.

User credentials are shared among the MYSEA Servers in the federation, allowing an authenticated user to gain access to data and services on different MYSEA Servers without having to log in at each MYSEA Server.

B. *Special Purpose Trustworthy Components*

The Trusted Path Extension (TPE) and the Trusted Channel Module (TCM) are specialized devices that, under the direction of the MYSEA Servers, either block or pass data and service requests to the MYSEA Servers.

The TPE is conjoined with an untrusted client workstation and acts as a gate keeper between the workstation and the MYSEA cloud. It provides an interface that allows a user to establish a trusted path between the user and the MYSEA Servers. Once the trusted path is established, the user can securely log in and negotiate a session level that is constrained by the user's clearance. The TPE also manages IPsec tunnels between itself and the MYSEA Servers whose cryptographic attributes are dynamically configured based on the TPE's unique identification and the negotiated session level. After a user session is initiated, the TPE forwards service requests from the workstation to the MYSEA Servers via these tunnels.

While the TPE controls network connectivity to the MLS LAN, the TCM serves as a multiplexer that labels incoming network traffic from single-level services and remote networks, and forwards the labeled data to the MYSEA Server over an MLS network interface. The TCM also uses IPsec to protect data transmitted between itself and the MYSEA Servers. Unlike the IPsec tunnels on the MLS LAN, the cryptographic attributes of the IPsec tunnels on a TCM's MLS interface are determined based on the security attributes of the single-level components. The use of the TCM in a particular operational environment depends on the number of physical network interfaces available in the federation. The TCM is used when it is necessary for multiple single-level networks to share the same physical network interfaces.

C. *Commodity Clients and Servers*

Users on the MLS LAN may interact with different MLS servers in the MYSEA cloud via stateless client workstations. These workstations are commodity platforms running commercial or open source operating systems and popular software applications, resulting in lower operating cost. A TPE is associated with each workstation. The MYSEA Server interprets the actions of each workstation to be at the user's negotiated session level, and the workstation's system state must be purged at the end of each session to prevent residual data leakage. To the extent possible, all user data objects and related metadata are stored on MYSEA Servers, not on the workstations; the exceptions are addressed by the workstation's power-cycle protocol.

Single-level services also run on commodity platforms. They include application services hosted on single-level servers in the local MLS enclave, and application services hosted on servers in remote single-level networks. Single-level servers in local MLS enclave provide application services to both local and remote clients operating at the same classification level. The read-down capability will be provided to users on single-level networks in a later implementation phase.

D. *Security Features*

The high-level design goals for MYSEA dictate that the architecture will 1) provide a distributed collaborative user environment that can interoperate with different platforms and be expandable in both functionality and performance, 2) adapt during run-time to support different threat conditions, e.g., normal, crisis, 3) require minimal user training and 4) scale to support up to 100 user workstations. MYSEA also supports the following security features:

- Secure connections to classified networks
- Centralized security management

- Use of adaptive security techniques to provide dynamic security services
- True multilevel access to data at multiple levels of security using a single commodity workstation
- Integration of multilevel security with existing sensitive networks using high assurance trusted communication channels
- Secure single sign-on across multiple MLS servers
- Server replication to support scalability
- IPv6 in a multilevel context
- Interoperability with the DoD PKI infrastructure
- High assurance trusted path and trusted channel techniques for managing access to the MLS cloud.

The next section discusses the major threats and assumptions pertaining to the current MYSEA implementation.

IV. Threats and Assumptions

The threat model for MYSEA includes both developmental threats and operational threats. Insider attacks can be mounted in all phases of a system's life cycle, e.g., during development or while in operation [38].

Unauthorized changes to a system's security mechanisms in any life cycle phase could adversely affect the system's ability to enforce its security policies. Our rigorous development life cycle management processes [39] address the threats of subversion and unintentional errors made by the development team.

It is assumed that organizational policies and operational procedures will be imposed to address exploitation by insiders or by unauthorized individuals with physical access to the networks, systems and facilities during deployment. Attacks in the supply chain and malicious insertion of unintended functionality during delivery for which mitigations involve various forms of tamper detection and prevention are outside of our threat model.

Operational threats include attacks on the network, malicious software and misbehaving users. Network attacks to the communication protocols within the MLS LAN or the MLS cloud can be passive (e.g., traffic monitoring and analysis) and active (e.g., deliberate circumvention of protection features). To counter some of these threats, MYSEA depends on IPsec to provide confidentiality and integrity protection, data origin authentication, anti-replay service, and access control based on cryptographic keys [31].

Another type of active attack involves malicious software (e.g., Trojan Horse) that attempts to either directly or indirectly gain unauthorized access to information by leveraging the user's own privilege [40]. This threat is mitigated in the MYSEA system by preventing the results of actions taken by a Trojan Horse from flowing to an object with less sensitivity. All resources on the MYSEA Servers are assigned security levels that are used by the underlying STOP security kernel to enforce policies for information confidentiality and integrity. The untrusted application protocol servers launched on behalf of the requesting workstations at the negotiated session level only have read-write access to resources at the session level and read-only access to resources at levels below the session level, effectively confining the actions of any Trojan Horse.

User or application misbehavior includes attempts by users at the client workstation or their application software to bypass the TPE. To mitigate these attacks, each TPE is required to register with the federation before any user actions are allowed, including the invocation of the trusted path.

V. Security Policies

MYSEA controls access to resources (e.g., data objects, network interfaces) using both mandatory access control (MAC) and discretionary access control (DAC). For MAC, MYSEA enforces lattice-based confidentiality and integrity policies [17, 18]. Information of different security classifications (e.g., TOP SECRET, SECRET NATO) is assigned different sensitivity labels which are used by the MYSEA Server to mediate access to data objects. This type of control is also called non-discretionary since processes cannot manipulate or bypass the policy rules. Under DAC policy, MYSEA access decisions are based on user identities and access permissions given to data objects by the users. Unlike with MAC, processes acting on behalf of users can, at their discretion, control by whom and how a resource is accessed.

In an MLS system, the enforcement of MAC and DAC policies must be supported by two accountability policies: Identification and Authentication (I&A) and Audit [14, 28]. For I&A, the MYSEA Server ensures that users are afforded a trusted communication path between the user and the MYSEA Server, and that the user's claimed identity and authentication credentials are validated before a user session is established. Regarding Audit, the MYSEA Server accounts for all users actions, either taken directly by the user (e.g., trusted path invocation) or by software acting on the user's behalf (e.g., a web server process). An audit trail of accesses is maintained and protected by the MYSEA Server.

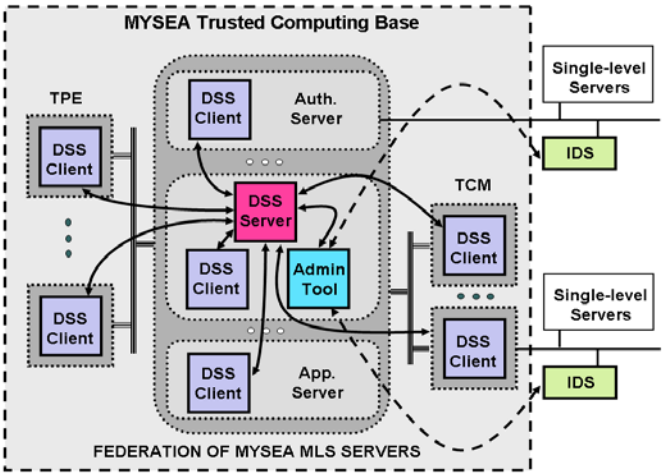

Figure 2. MYSEA Dynamic Security Services Framework

VI. Dynamic Security Services

To meet run-time adaptability objectives, MYSEA implements a service-based access control policy that restricts the client workstations to access services hosted on the MYSEA Server. The Dynamic Security Services (DSS) mechanisms support automatic revocation of access to application services (e.g., web browsing, email) and dynamic modulation of protection attributes of the IPsec tunnels in response to network conditions (e.g., INFOCON threat levels) and policy constraints on the user session (see Fig. 2).

The DSS design follows the standard policy management paradigm that includes a policy input point (PIP), a policy repository, a policy decision point (PDP) and one or more policy enforcement points (PEP) [41]. A push model is used for policy delivery, i.e., the PDP pushes policy rules to the PEPs based on a triggering event. The policy input point provides a human interface with which the system administrator can create, edit and promulgate policies. This component interacts with the policy repository and the PDP. The policy repository stores the PIP-sanctioned policies and provides those policies to the PDP when requested. The PDP retrieves policies from the policy repository and makes DSS control decisions on behalf of the PEPs which enforce device-specific DSS policy rules.

The DSS mechanism consists of the three elements. The *DSS Server* acts as a PDP and services DSS requests from the DSS Client. The *DSS Client* performs policy enforcement functions using an access control list in the form of IPsec rules and IKE security associations. The *DSS Administration Tool*, operating as a PIP, allows the administrator to manage the DSS policies to reflect the system operating modes and the enterprise security policy in force. The DSS Control Protocol, a custom TCP-based command-respond protocol, is used by these elements to communicate with each other (solid arrows in Fig. 2).

MYSEA employs external intrusion detection systems (IDS) on the single-level networks to monitor for suspicious network activity on those systems. IDS alerts generated by these systems are securely stored in single-level SQL databases on the MYSEA Server. The IDS alerts are used by the security analysts to determine network conditions and security disturbances, which may result in a DSS policy change. Automation of policy changes based on IDS alerts is a topic of future research.

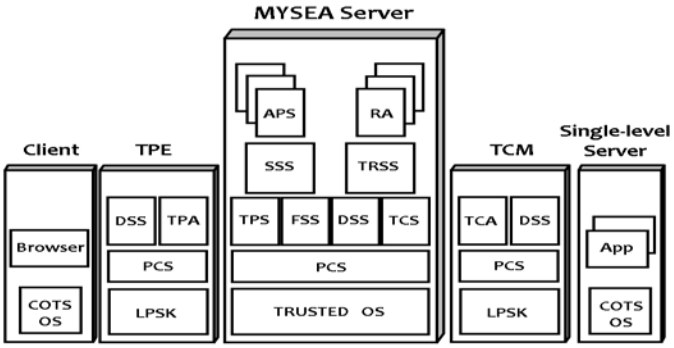

Figure 3. MYSEA Sofware Stack

VII. MYSEA Software Architecture

The MYSEA trusted computing base (TCB) is composed of the MYSEA Server, the TPE and the TCM. As illustrated in Fig. 3, a common core exists in all three TCB components: a high assurance operating environment (i.e., LPSK [29] and STOP OS [15]), the Protected Communications Service (PCS) and DSS. The PCS component implements IPsec tunnels to protect communications between the TPE and the MYSEA Server, and between the TCM and the Server. The DSS component performs dynamic service management functions that include changing IPsec configurations and negotiating session keys.

The Trusted Path Service (TPS) and Trusted Path Application (TPA) components are tightly coupled. The TPA on the TPE provides a human interface for the users to invoke the trusted path to login and establish a session. On the other hand, the TPS on the MYSEA Server performs user authentication and session negotiation functions in accordance with the I&A policy. Similarly, the Trusted Channel Service (TCS) and the Trusted Channel Application (TCA) components work together to ensure that traffic between a single-level network and the MYSEA Server are properly labeled at the classification level of the particular network.

Communications between MYSEA Servers in the federation is handled by the Federated Security Service (FSS). This component establishes an FSS session between two MYSEA Servers so that they can exchange federation management and single sign-on information. The FSS also performs federation-wide initialization and cleanup functions. The Secure Session Service (SSS) and Trusted Remote Session Service (TRSS) are trusted components. The SSS manages application requests from the client workstations and other MYSEA Servers in the federation while the TRSS manages network requests from remotely-executed applications. These components have special privileges that allow them access to resources of different security levels.

The Application Protocol Server (APS) and Remote Application (RA) components are not trusted and run at the security level of the user session from which they are invoked. Each APS is a TCP/IP protocol handler that is modified to provide cross-domain functionality, i.e., read-down. MYSEA currently supports SMTP, IMAP and HTTP protocols which together afford users the ability to gather information (web browsing), collaborate (wiki and WebDAV) and communicate (email and webmail). The RA is a modified client-side application process running on the MYSEA Server that uses MYSEA TCB interfaces to communicate with a server-side application running on a MYSEA Server or on an external single-level server. A TFTP client program accessing a TFTP server on the Internet (e.g., to download certain data) is an example of accessing an external server.

VIII. Discussion

The current MYSEA design supports many characteristics associated with cloud computing [42], including *broad network access* (MLS-constrained capabilities are accessible via standard web-based mechanisms by heterogeneous thick and thin clients), *resource pooling* (MLS computing resources, e.g.,

MLS servers, can be dynamically combined to support users in different domains with different computing demands), and *measured services* (DSS mechanisms can be viewed as a form of resource metering capability that can be utilized for service control and resource optimization). MYSEA's "MLS Cloud" is presently oriented towards the *Cloud Software as a Service* (SaaS) model, where the users (consumers) can access cloud applications from a web-based interface (e.g., webmail) but have little control over how those applications are managed. In terms of ownership, administrative domain, and availability to a larger community, the MYSEA cloud could be deployed as a *private cloud*, a *community cloud*, or a *hybrid cloud*.

MYSEA does not yet provide the ability to provision computing resources, such as servers and applications, unilaterally by the users (*on-demand self-service*) or automatically by the cloud (*rapid elasticity*). The future design of the "MLS Cloud" will explore the impact of providing these services; for example, does the dynamic resource provisioning associated with "on-demand self-service" create potential covert channels? Supporting the *Cloud Platform as a Service* (PaaS) model is another potential enhancement that would allow the users to leverage MYSEA's high assurance computing base, resulting in lower operating cost. With PaaS, the users can create and deploy their own applications using the development tools and programming interfaces provided by the (MYSEA) cloud.

IX. Related Work

Hinke suggested the idea of a high assurance server to provide a locus of multilevel secure control to single level clients [43]. Unlike in MYSEA's design, clients are restricted to a single level throughout their lifetime. Rushby and Randell [44] 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. Over and above this basic functionality, MYSEA presents 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.

Non-distributed approaches to support access to multilevel data via COTS applications have been proposed [45, 46, 47]. Purple Penelope has limited assurance, as it runs as a user-level application wrapping Windows NT, and it does not support a modifiable session level. The other systems rely on an underlying reference validation mechanism to control access to multilevel data. The MYSEA project extends certain concepts from these projects to a distributed environment.

Replication architectures [48] provide a simple technique to achieve near-term multilevel security by copying all information at low security levels to all dominating levels. MYSEA rejects this approach because, when considering many documents or security levels, replication scales so poorly as to be infeasible.

The Naval Research Laboratory (NRL) Network Pump [49] is a network guard that has been proposed as part of a larger network architecture connecting subnets at different sensitivity levels, resulting in a multiple single-level (MSL) network [50]. Starlight [51] was designed to support logically separate single-level workstations connected by an MLS-aware switch to data management subsystems at different (single) levels. The capital and administrative costs of separately maintained single-level LANs is a drawback that MYSEA avoids.

Regarding QoSS, there have been several efforts in this direction. A quality of protection parameter is provided in the GSSAPI specification [52]. This parameter manages the level of protection provided to a message communication stream by an underlying security mechanism (or service). Another approach is that of Schneck and Schwan [53], which discusses variable packet authentication rates with respect to the management of system performance.

X. Conclusions

Cloud computing promotes agility, scalability, collaboration, and sharing of resources across domains/organizations but inherits the same security risks associated with any distributed system handling high-value data and resources. MYSEA integrates support for cloud computing functionality with the strong security properties provided by a high-assurance multi-domain system. MYSEA's architectural elements consist of a federation of highly trustworthy MLS servers, a set of special purpose authentication components and commodity client workstations. MYSEA's security features include strong cross-domain access controls, protection of system assets (data and services) with different security classifications, resource isolation, service replication and dynamic control of QoSS. MYSEA also hosts MLS-constrained collaborative application services that are accessible via standard protocols (e.g., HTTP, SMTP/IMAP, SIP-based VoIP).

References

[1] CNSS Instruction No. 4009, "National information assurance (IA) glossary," Committee on National Security Systems, Revised June 2006.

[2] M. Bailey, "The unified cross domain management office: bridging security domains and cultures," *CrossTalk magazine*, vol. 21, no. 7, pp. 21–23, July 2007.

[3] P. Gutmann and I. Grigg, "Security usability," *IEEE Security and Privacy*, vol. 3, no. 4, pp. 56–58, July/August 2005.

[4] C. E. Irvine, T. D. Nguyen, D. J. Shifflett, T. E. Levin, J. Khosalim, C. Prince, P. C. Clark, and M. Gondree, "MYSEA: the Monterey security architecture," Proc. of the Workshop on Scalable Trusted Computing (ACM STC), Conference on Computer and Communications Security (CCS), Association for Computing Machinery (ACM), Chicago, Illinois, November 2009, pp. 39–48.

[5] 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).

[6] W. H. Ware, "Security and privacy in computer systems.," AFIPS '67 (Spring): Proc. of the April 18-20, 1967, Spring Joint Computer Conference. Atlantic City, New Jersey, pp. 279–282.

[7] C. Weissman, "Security controls in the ADEPT-50 time-sharing system," AFIPS '69 (Fall): Proc. of the November 18-20, 1969, Fall Joint Computer Conference, Las Vegas, Nevada. pp. 119–133.

[8] W. L. Schiller, "The design and specification of a security kernel for the PDP-11/45, MTR-2934," MITRE Technical Report MTR-2934, March 1995, The MITRE Corporation, Bedford, MA 01730.

[9] L. J. Fraim, "Scomp: a solution to the multilevel security problem," *IEEE Computer*, vol. 16, no. 7, July 1983. pp.26–34.

[10] Department of Defense Computer Security Center, "Final Evaluation of SCOMP, Secure Communications Processor, STOP Release 2.1," CSC-EPL-85-001, September 23,1985.

[11] R. Schell, T. Tao and M. Heckman, "Designing the GEMSOS security kernel for security and performance," Proc. of the 8th National Computer Security Conference, September 1985.

[12] National Computer Security Center, "Final evaluation report: Boeing space and defense group, MLS LAN secure network server system," 28 August 1991.

[13] National Computer Security Center, "Final evaluation report of Wang Government Services, incorporated XTS-300," CSC-EPL-92/003C-Evaluation No. 21-92, 27 April 1999.

[14] National Computer Security Center, "Department of Defense trusted computer system evaluation criteria," DoD 5200.28-STD, December 1985.

[15] BAE Systems Information Technology, LLC, "Security target, version 1.22 for XTS-400, version 6.4.U4," June 2008.

[16] Common Criteria Project Sponsoring Organizations, "Common Criteria for information technology security evaluation," Parts 1–3, Version 2.3, August 2005.

[17] D. E. Bell and L. LaPadula, "Secure computer system: unified exposition and Multics interpretation," Technical Report ESD-TR-75-306, The MITRE Corporation, Hanscom AFB, MA, 1975.

[18] K. J. Biba, "Integrity considerations for secure computer systems," Tech. Report ESD-TR-76-372, The MITRE Corporation, 1977.

[19] J. Rushby, "Design and verification of secure systems," ACM Operating Systems Review, vol. 15, no. 5, December 1981.

[20] J. Rushby, "Partitioning in avionics architectures: requirements, mechanisms, and assurance," Technical Report, Computer Science Laboratory, SRI International, March 1999.

[21] Green Hills Software, "INTEGRITY-178B separation kernel security target," Version 1.0, May 2008.

[22] IBM, "Common Criteria for information technology security evaluation public version of the security target for PR/SM for the IBM system z10 EC," Version 7.7.2, October 2008.

[23] Trusted Information Systems, Inc., "A proposed interpretation of the TCSEC for virtual machine monitor architectures, Vol. 1: strict separation," Draft of 1 May 1990, Trusted Information Systems, Inc., Glenwood, MD. (Unpublished)

[24] National Security Agency, "U.S. government protection profile for separation kernels in environments requiring high robustness," Version 1.03, 29 June 2007.

[25] T. E. Levin, C. E. Irvine and T. D. Nguyen, "Least privilege in separation kernels," Proc. of International Conference on Security and Cryptography, Setubal, Portugal, August 2006, pp. 355–362.

[26] J. H. Saltzer and M. D. Schroeder, "The protection of information in operating systems," Proc. of the IEEE, vol. 63, no. 9, pp. 1278–1308, September 1975.

[27] National Institute of Standards and Technology, "Personal identity verification (PIV) of federal employees and contractors," Federal Information Processing Standards Publication 200-1 (FIPS PUB 201-1), March 2006.

[28] Common Criteria Project Sponsoring Organizations, "Common Criteria for information technology security evaluation, parts 2: security functional component", Version 3.1 Revision 3, CCMB-2009-07-002, July 2009.

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

[30] C. E. Irvine, T. E. Levin, P. C. Clark and T. D. Nguyen, "A security architecture for transient trust," Proc. of Computer Security Architecture Workshop, Fairfax, Virginia, USA, October 2008.

[31] S. Kent and K. Seo, "Security architecture for the Internet Protocol," Request for Comments Number 4301 (RFC 4301), The Internet Society, December 2005.

[32] S. Kent, "IP authentication header," Request for Comments No. 4302 (RFC 4302), The Internet Society, December 2005.

[33] S. Kent, "IP encapsulating security payload (ESP)," Request for Comments No. 4303 (RFC 4303), The Internet Society, December 2005.

[34] C. Kaufman (Editor), "The Internet key exchange (IKEv2) protocol," Request for Comments No. 4306 (RFC 4306), The Internet Society, December 2005.

[35] C. Aurrecoechea, A. Campbell, and L. Hauw, "A survey of quality of service architectures," Multimedia Systems, vol. 6, no. 3, pp. 138–151, 1996.

[36] 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," Proc. of 11th IPPS/SPDP'99 Workshops, pp. 1316–1326, Berlin, April 1999.

[37] T. E. Levin, C. E. Irvine and E. Spyropoulou, E., "Quality of security service: adaptive security," Handbook of Information Security, Vol. 3, pp. 1016–1025, ed. H. Bidgoli, John Wiley and Sons, Hoboken, NJ, 2006.

[38] P. Myers, "Subversion: the neglected aspect of computer security," Master's thesis, Naval Postgraduate School, Monterey, CA, 1980.

[39] 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.

[40] B. Lampson, "A note on the confinement problem," Communications of the ACM, vol. 16, no. 10, pp 613–615, 1973.

[41] A. Westerinen et al., "Terminology for policy-based management," Request for Comments No. 3198 (RFC 3198), The Internet Society, November. 2001.

[42] P. Mell and T. Grance, "NIST definition of cloud computing," Version 15, National Institute of Standards and Technology, Information Technology Laboratory, October 2009. URL: http://csrc.nist.gov/groups/SNS/cloud-computing/cloud-def-v15.doc. Last accessed: April 14, 2010.

[43] T. Hinke, "The trusted approach to multilevel security," Proc. of Computer Security Applications Conference, pp. 335–341, December 1990.

[44] J. Rushby and B. Randell, "A distributed secure system," Proc. of IEEE Symposium on Security and Privacy," pp. 127–135, May 1983.

[45] D. E. Denning, T. F. Lunt, R. R. Schell, W. Shockley and M. Heckman., "The SeaView security model," Proc. of IEEE Symposium on Security and Privacy, pp. 218–233, April 1988.

[46] T. F. Lunt, R. R. Schell, W. Shockley, M. Heckman and D. Warren, "A near-term design for the SeaView multilevel database system," Proc. of IEEE Symposium on Security and Privacy, pp. 234–244, 1988.

[47] B. Pomeroy and S. Weisman, "Private desktops and shared store," Proc. of 14th Computer Security Applications Conference, pp. 190–200, Phoenix, AZ, December 1998.

[48] J. Froscher, M. Kang, J. McDermott, O. Costich and C. E. Landwehr, "A practical approach to high assurance multilevel secure computing service," Proc. of Computer Security Applications Conference, pp. 2–11, Orlando, FL, December 1994.

[49] M. H. Kang, J. N. Froscher and B. J. Eppinger, "Towards an infrastructure for MLS distributed computing," Proc. of 14th Annual Computer Security Applications Conference, pp. 91–100, Phoenix, AZ, December 1998.

[50] M. H. Kang and I. Moskowitz, "Design and assurance strategy for the NRL pump," IEEE Computer, vol. 31, no. 4, pp. 56–64, April 1998.

[51] M. Anderson, C. North, J. Griffin, R. Milner, J. Yesberg and K. Yiu, "Starlight: interactive link," Proc. of 12th Computer Security Applications Conf., San Diego, CA, December 1996.

[52] J. Linn, "Generic security service application program interface, version 2, update 1," Request for Comments No. 2743 (RFC 2743), The Internet Society, January 2000.

[53] P. A. Schneck and K. Schwann, "Dynamic authentication for high-performance networked applications," Tech. Report GIT-CC-98-08, Georgia Institute of Technology College of Computing, 1998.