Journal of Computer Science 3 (9): 726-735, 2007
ISSN 1549-3636

Policy-Based Security for Wireless Components in High Assurance Computer Systems

Luay A. Wahsheh and Jim Alves-Foss
Center for Secure and Dependable Systems, University of Idaho, P. O. 441008
Moscow, Idaho 83844-1008, USA

Abstract: To enable the growth of wireless networks in high assurance computer systems, it is essential to establish a security engineering methodology that provides system security managers with a procedural engineering process to develop computer security policies. Our research demonstrates how wireless communication technology is deployed using the Multiple Independent Levels of Security (MILS) architecture for high assurance computer system design of security and safety-critical multi-enclave systems to provide a framework for supporting the enforcement of diverse security multi-policies. The established wireless inter-enclave multi-policy paradigm manages multiple wireless security policies within heterogeneous systems. Applying the policy refinement rules presented in this work for a security enforcement procedure of an application system will reduce the proof effort for secure components.

Keywords: High assurance architecture, policy refinement, security policy, wireless.

INTRODUCTION

High assurance computer systems are those that require convincing evidence that the system adequately addresses critical properties such as security and safety objectives[8]. They are used in environments where failure can cause security breaches or even the loss of life. Examples include avionics, weapons controls, intelligence gathering, and life-support systems. Before such systems can be deployed, there must exist convincing evidence that they support critical safety, as well as security, properties.

Security in high assurance computer systems involves protecting systems' entities from unauthorized access. We use the term entity to refer to any source or destination through which information can flow (e.g., user, subject, object, file, printer). Several issues have to be addressed in order to have systems function in a secure manner, including authorization, authentication, and software and hardware correctness. Our work focuses on security policies in relation to wireless communication. In this paper, we use the following terms: security enclave (coalition) to refer to a logical boundary for a group of entities that have the same security level; and message to refer to any data that has been encoded for transmission to or received from an entity (e.g., a method invocation, a response to a request, a program, passing a variable, a network packet). The transmission mechanism can utilize shared memory, zero-copy message transport, kernel supported transport, TCP/IP, and so forth.

In the computer security literature, the term policy has been used in a variety of ways. Policies can be a set of rules to manage resources (e.g., actions based on a certain event(s)) or definite goals to determine present and future decisions; we provided a detailed discussion of the meaning of policy in our earlier work[17]. Broadly speaking, a computer policy should address security issues: CIA (Confidentiality, Integrity, Availability). Although not a requirement for the work described in this paper, our work focuses on MILS (see the MILS Architecture Section). Security policies in MILS can be multi-level (e.g., based on security classification: Top Secret, Secret, Confidential, Unclassified) and contain mandatory components that specify rules that guarantee only authorized message transmission between entities by imposing constraints on the actions (operations) of these entities. In addition, MILS can support other types of policies such as RBAC, DAC, corporate policies, and so forth.

The issues of wireless communication and demand for mobility have recently been receiving more attention. The field of wireless security policies and policy engineering is relatively new. There exists various research work in the literature that discusses security policies. However, very little of this work discusses enforcing diverse multi-policies in high

Corresponding Author: Jim Alves-Foss, Director, Center for Secure and Dependable Systems, University of Idaho, P. O. Box 441008, Moscow, Idaho 83844-1008, USA. Tel.: +1-208-885-4114, Fax: +1-208-885-6840

assurance multi-enclave systems in the context of wireless communication technology.

This paper outlines a novel wireless inter-enclave multi-policy technique that provides system security managers with a framework for supporting the enforcement of diverse security multi-policies in high assurance computer systems. We present an architecture that provides a basis for the support of multiple policies, both individually and in composition. The architecture provides for a procedural mapping of high-level system security policies into low-level implementation mechanisms that can be verified to enforce the policies.

Although security plays a major role in the design of software systems, it is still not considered an explicit part of the development process. Security requirements are usually added to an already existing system. As a result, this leads to numerous problems with the overall security design. Policies should be taken into account early on in the development process. The problems and techniques that this research presents are significant because wireless security policies play an important role in the success of a secure wireless environment.

WIRELESS SECURITY

Because wireless communication provides an increase in connectivity, it also creates an increase in security vulnerabilities. In wireless networks, the communication medium (air) is a major source of vulnerability that jeopardizes security. While wireless networks provide increased mobility for entities, they provide easier access for attackers. To access traffic in a traditional wired network, an entity has to be physically connected to the network.

What is Wireless?: Wireless is a technology that uses radio frequency to allow transmission of information over electromagnetic waves between communicating entities without establishing a physical connection between them via cables. Wireless technology is emerging as a significant medium through which signals can travel as a means for communication. A wireless network is an addition to an existing wired network foundation. It provides entities with access to the Internet and/or network resources without being physically connected via cables. A wireless network can be established using two design structures: infrastructure or ad hoc. An infrastructure network uses a wireless access point to transmit messages between entities. A wireless access point is a device that is usually connected to a wired network and can send and receive information between wireless and wired devices. It acts as an Ethernet bridge between a wireless entity and a wired network. An ad hoc network allows entities to directly connect to one another without having to go through an access point.

Wireless Security Challenges: Wireless networks broadcast data into the air and any wireless entity within range can monitor traffic. However, we propose using wireless technology in a MILS system mainly due to free movement convenience and ease of implementation. Its benefits include the following:

- Mobility: entities that are on the move (changing position all the time) can connect to the network and have access to information regardless of their location.
- Ease of installation procedures and implementation: no laying out of cables is needed, which allows for fast installation.
- Installation cost: in dynamic environments where there are frequent changes, setting up a wireless environment will save costs of laying new cables.
- Flexibility: once the infrastructure has been set, new entities can be added to the system without the need for any extra hardware or software and without affecting other existing entities. In addition, there are situations where cable is not possible (e.g., air vehicles).

With wireless benefits come certain shortcomings, including:

- Security risks: some security challenges of wireless networks include lack of or weak encryption and user authentication. Computer attacks enable intruders not only to have access to the information that is being transmitted over the air, but also the ability to transmit messages on the wired network. Wireless networks are open to several attacks, including attacks on network confidentiality, integrity, and availability[13]. Attacks on confidentiality include eavesdropping, entity authentication compromise, and encryption key compromise. Attacks on integrity involve unauthorized modification of information (e.g., man-in-the-middle attack, session hijacking). Attacks on availability involve preventing entities from using a resource (e.g., denial-of-service attack).
- Bandwidth and speed overhead: wireless devices are restricted to function in some electromagnetic bandwidth; lower data transmission rate is caused by lower network bandwidth and noise. As a result, the speed of wireless networks is less than that of wired networks and in network congestion situations, entities will face delays due to network performance.
- Radio wave interference: due to severe weather conditions, wireless signals may be prevented from being transmitted properly. Also, the signal strength may be limited due to geographic obstacles (e.g., mountain, bridge).

Since wireless networks are more vulnerable to attacks, it is crucial to implement measures to prevent such attacks and secure a wireless network. Such measures include firewalls, end-to-end encryption of all

information being transmitted, user authentication to prevent unauthorized users, virus protection, and prevention of unauthorized (rogue) access points. In addition, security audits should constantly be implemented to ensure the security of wired and wireless networks. Logging wireless and wired events enables system security managers to identify scenarios where attackers might be attempting to compromise a network.

Wireless Communication Security Policies: Policies for wired and wireless networks are crucial elements of systems' security. In MILS, security policies are designed not only to guide information access, but also to control conflicts and cooperation of security policies of different security enclaves[17]. We strongly believe that no enforcement of security standards can be effectively made without the support of security policies.

One important issue policy developers should keep in mind when designing security policies is that policies have to be flexible enough to evolve with new wireless environments. According to Verma[16], in order to be effective (flexible), policies need to meet certain requirements: policies must be easily specified and understood by human operators, precisely defined and enforced, compatible with the capabilities of the network element where they may be enforced, and consistent to avoid conflicts and ambiguous decision-making.

Wireless security policies play an important role in the success of a secure wireless environment. As a result, careful design, implementation, and enforcement of security policies are crucial in reducing security vulnerability while at the same time maximizing network performance. In addition, routine audits should be done to monitor policy compliance and wireless usage so that all activities can be traced back to an entity. Despite the security risks associated with wireless technology, the security of wireless networks can be increased by developing and implementing a comprehensive security policy along with the use of new wireless technology devices and enforcement mechanisms.

MILS ARCHITECTURE

In the past, secure systems were designed with the concept of a security kernel and a Trusted Computing Base (TCB)[6]. The key concept behind this approach is that the security decisions and security enforcement mechanisms are an integral part of the TCB. Following this design paradigm, development teams found that more and more of their system's functionality was being included in the TCB. Once this occurred, the evaluation of the system's security became unmaintainable.

Traditionally, the military model of a secure operating system includes the concept of multi-level security (MLS). The idea behind this concept is that the system will be processing data items that are classified at different levels of security, and the information flow security policy that prevents the transfer of high-level classified information into unauthorized objects must be preserved. The MLS concept has applications outside the military, including communications within critical infrastructures and safety-critical real-time control systems. Therefore, we define an MLS system as one that processes and outputs data at multiple classification levels. Classic security models, such as the Bell-LaPadula model[5], have been used to specify the secure behavior of MLS systems.

The Need for MILS: MILS is a joint research effort between academia, industry, and government led by the United States Air Force Research Laboratory with stakeholder input from many participants, including the Air Force, Army, Navy, National Security Agency, Boeing, Lockheed Martin, Objective Interface Systems, Green Hills Software, Lynux Works, Wind River, General Dynamics, Raytheon, Rockwell Collins, MITRE, and the University of Idaho[1, 2, 7].

The MILS architecture is created to simplify the process of the specification, design, and analysis of high assurance computer systems[19]. This approach is based on the concept of separation, as introduced by Rushby[14]. Through separation, we can develop a hierarchy of security services where each level uses the security services of a lower level or peer entities to provide a new security functionality that can be used by higher levels. Effectively, the operating system and middleware become partners with application level entities to enforce application-specific security policies. Limiting the scope and complexity of the security mechanisms provides us with manageable, and more importantly, evaluatable implementations. A MILS system isolates processes into partitions, which define a collection of data objects, code, and system resources. Partitions are defined by the kernel's configuration and can be evaluated separately. This divide-and-conquer approach will reduce the proof effort for secure systems.

What is needed is a complete system architecture that partitions system functionality into manageable units. The MILS architecture does precisely that, it works by partitioning programs, their data, and their

communications. A traditional deployment would consist of a Separation Kernel (SK), Real-Time Operating System (RTOS), CORBA middleware, a GIOP Guard, a MILS Message Router (MMR), and a Partitioning Communications System (PCS).

Definition 1: A guard is a trusted (satisfies security requirements) computing component that enforces policies associated with certain types of communication channels. The type of messages it can sanitize is unique (e.g., GIOP, HTML, TCP/IP).

Definition 2: An MMR is a trusted computing component that enforces both edge policies and inter-partition communication. If messages between partitions are of a specific type, then the MMR routes these messages to appropriate guards or rejects them.

Definition 3: A PCS is a trusted computing component that enforces inter-processor communication.

Several MILS benefits are appealing to government departments and agencies, including the military and defense systems. Information, no matter what domain it belongs to, can exist on the same distributed system while preserving separation. The need for MLS systems has increased with the interconnection of multiple systems into a GIG (Global Information Grid). The MILS architecture has several advantages, including[9]:

- The difficulty of certification of MLS systems is resolved by separating the security mechanisms and concerns into manageable components. This provides an increase in the security and safety of systems.
- Hardware is reduced since a system can be built on the separation kernel where physically isolated processors are not required. This provides space and power reduction.
- A single physical processor can host multiple applications at different security levels. This provides easier management of information between different entities.
- Real-time performance can be supported in a securely partitioned system.

MILS Layers: The MILS architecture is designed and implemented in layers. MILS is divided into three layers, which consists of a separation kernel (and hardware: processor, physical memory, assigned devices), middleware (and operating system services), and application. Each policy enforces security at a given layer and provides secure services for the layer that is immediately above it. Some partitions will be designated multi-single level secure (MSLS), consisting of a single data classification, while others will be multi-level secure supporting several data classifications[2]. Figure 1 shows the MILS layered architecture using Top Secret (TS) and Secret (S) applications. Notice that the partitions could be running different operating systems (OS1, OS2, OS3, OSn) or the same operating system.

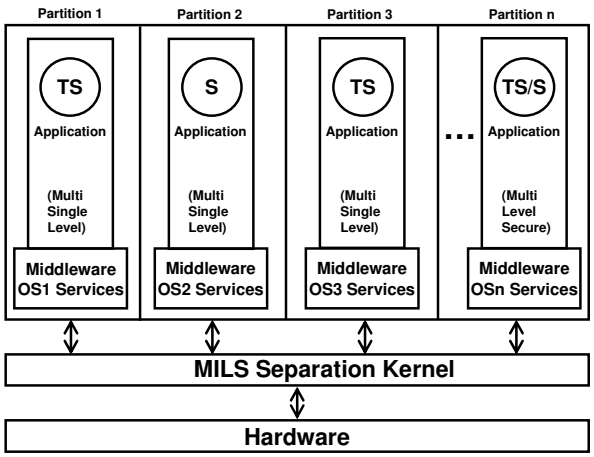

Fig. 1: MILS layered architecture.

The SK segregates entities and their resources into isolated execution spaces (partitions). The SK divides a host processor into multiple partitions that are logically separated in space and time. Each partition appears to have its own dedicated processor (virtual machine). The separation kernel partitions space into different memory areas for each process to access. The SK partitions time into intervals that are allocated to each entity. Time is allocated based on time-slice scheduling (static) or preemptive priorities (dynamic).

Entities running in different partitions cannot communicate unless explicitly permitted by the separation kernel. However, the SK only enforces communication at the message level. If needed, we can place the MMR in a separate partition to enforce finer granularity policies. If messages between partitions are of a specific type, then the MMR routes these messages to appropriate guards or rejects them (possibly sending a specified response). If a guard determines that the content of a message is not in compliance with the information flow policy, the guard will notify the MMR which will then disallow the communication attempt or take action based on the security policy.

Separation Kernel Layer: The foundational component of MILS is the separation kernel which is the lowest level layer. It creates partitions and monitors

change of control between them. The SK is a small software component (typically not more than 6,000 lines of code) that is trusted to guarantee separation of time and space partitioning. MILS SK is based on enforcing data isolation, information flow, periods processing, and damage limitation policies. Data isolation protects data segments within partitioned applications from being read or corrupted by unauthorized entities or applications. Information flow ensures that only authorized communication between partitioned applications can occur. Periods processing ensures that when the execution between partitioned applications is being switched, shared resources (e.g., processor registers) are sanitized (no information from one partition can be read by another). Damage limitation guarantees that failure caused by damages in one partitioned application does not compromise the continued processing of other applications.

Middleware Layer: The MILS middleware contains mediation and communication components that are responsible for controlling messages between entities. It could include a secure version of real-time CORBA and other operating system services that are excluded from the SK, such as device drivers and network services[2]. It also includes a trusted PCS component that extends the separation that the SK provides to include communication between different distributed systems. The middleware may also include a publish-subscribe DDS and DBMS libraries. The middleware can reside in the same partition that an application uses or in separate partitions.

Application Layer: At this layer, users run their applications that get assigned to different partitions. Within the MILS architecture, application layer entities are provided with the mechanisms to control, manage, and enforce their own application security policies in a manner that ensures that the enforcement mechanisms are NEAT (Non-bypassable, Evaluatable, Always invoked, Tamperproof). Non-bypassable means that the mechanisms cannot be avoided even through the use of lower-level functions. Evaluatable means that the mechanisms are simple enough to be analyzed and mathematically verified. Always invoked means that the mechanisms are invoked every time an action occurs (they must mediate every access). Tamperproof means that the mechanisms cannot be changed by unauthorized entities.

WIRELESS POLICY ENFORCEMENT

We defined IEMP (Inter-Enclave Multi-Policy) and PEG (Policy Enforcement Graphs) in our earlier work[17, 18]. In this paper, we extend MILS multi-policies to include wireless communication capabilities using the PEG approach that provides guidelines for wireless policies in a way that the security mechanisms are still NEAT across the entire network. By doing this, we obtain a communication architecture that allows the separation kernel, middleware, and applications to share the responsibility of creating a highly secure distributed system.

Wireless Issues: MILS supports security multi-policies that define what is allowed. The goal of wireless policies is to prevent unauthorized access to wireless broadcasting of sensitive data. With increasing complexity of computer networks and resources to be managed, the PEG approach allows a system architect to develop a secure wireless network that protects information and prevents unauthorized data access. System security managers will have a centralized access and information flow management over the wired and wireless network.

The scope of MILS wireless policies covers all wireless communication devices that are connected to the MILS network and are developed for securing wireless devices and transmissions. Wireless policies address the following issues:

- Using end-to-end data encryption on wireless systems and defining the encryption requirements of all wireless connections.
- Using end-to-end user authentication on wireless systems and defining the authentication requirements of all wireless connections.
- Identifying the legitimate communication source and destination channels on the network.
- Stating who has the responsibility of maintaining the wireless system (the system security manager).
- Preventing deploying a wireless device without permission from the system security manager. The manager reviews the device to ensure that it has a suitable level of security before updating the wireless policy with the appropriate handling instructions.
- Defining the equipment and protocols that will be used. All wireless equipment is required to be registered with the system security manager.
- Identifying the legitimate wireless access points.
- Keeping track of the location of entities using GPS (Global Positioning System) and control the location from which a wireless entity can use the system's access points. For example, Malaney[10] used the position of the requesting entity in order to mediate entity authentication. He presented a security system that uses GPS and signals generated by the wireless

entities to determine the location of authorized entities.

- Security policies in MILS cover wireless and wired networks. If a security attack occurs on the wireless network, this attack will not have a tremendous effect on the wired network. A network access point is used to help facilitate wireless and wired communication. Similar to wired networks, wireless networks provide entities with a communication platform that allows entities to connect with one another and share resources. Unlike wired networks, wireless networks use air to transmit data. In order to have more coverage area, communication systems usually set up more access points. In order to reduce exposure to security vulnerability when providing wireless communication through access points, the number of access points in the wireless network should be minimized.

MILS Wireless Architecture: Information access controls are the mechanisms involved in the mediation of every request to resources and data maintained by a system. Based on the security policy, they determine whether the request should be granted or denied. This mediation must be performed by a trusted component, the MILS Guarded Communications System (GCS).

GCS: Policies are enforced in MILS using mechanisms built into the kernel and middleware security components. The GCS is a logical subset of the middleware that consists of libraries and stubs in user partitions as well as individual enforcement mechanisms in separate partitions (e.g., guards, downgraders, encryption engines, message routers). In the example shown in Fig. 2, the GCS consists of the following trusted components: the PCS, network protocols, MMR, and guard. The separation kernel enforces compliance to information flow policies using the GCS component. This component can be verified independently and therefore can be used to mediate message passing between partitions.

The advantage of using the GCS is that the system does not have to trust the applications to conform to security policies. The GCS will enforce these policies. Thus, it is possible to have a secure MILS system while running untrusted applications within partitions. This is because the SK prevents any other possible partition communication. The SK, in conjunction with the GCS, enforces MLS policies.

The GCS makes access decisions in individual enclaves or between different enclaves using a policy database that stores the policies that the GCS will need. The system security manager has the authority to specify security policies that are enforced by the system. Auditing can be performed for entity requests; a request can be logged as a trace operation which will be used for analysis of activities in the system. Different policies can all exist in one policy database. If the invoking entity is allowed to access another entity, then access is granted; otherwise it is denied. The GCS is responsible for enforcing and monitoring the individual security policies and the multi-policies that are related to entities involved in the access.

The GCS is the collection of policy enforcement mechanisms that mediate message transmission between entities. Once an entity makes a request to pass information, the request will trigger the policies that are related to the requesting entity. The GCS receives the request and identifies the policies that have been triggered. The GCS is separate from the policy database, which makes the system flexible and simple; the system security manager will be able to change policies without modifying the enforcement mechanism.

To avoid unauthorized disclosure of information, it is necessary that messages are properly labeled. The GCS requires a label on all outgoing messages across the network. Messages labeled by the GCS are considered MILS compliant (they conform to the MILS architecture). MILS non-compliant messages are not labeled by the GCS and are sent from legacy (non-MILS) components. When the GCS receives such legacy messages, it validates the message and then queries the policy database for further information. Based on such information, if the MILS non-compliant message is given access permission, then the GCS will properly label it so it becomes MILS compliant. The specified label in the header of the message should help other MILS components (e.g., MMR, guards) to identify the message type and therefore support the communicating message.

The GCS is consistent and complete. It is consistent because an entity request is either accepted or denied but not both. This is due to the conflict resolution techniques that force the GCS to make a decision. The GCS is complete because for each entity request, there is a result (the access being accepted or denied).

PCS: The main security function of the PCS is to extend the single processor security policy enforcement provided by the MILS separation kernel to a distributed computing environment[12]. The MILS PCS is a middleware component that consists of hardware and software. The PCS functions as a communication interface that maintains secure communications

between entities running in separate partitions on different processors while enforcing security policies on those communications. It reduces the cost of designing, evaluating, and deploying highly secure systems.

The PCS restricts the use of channel numbers for use by particular entities. It mediates interactions between entities via channels according to two security policies: the channel connectivity policy and the resource management policy[12]. The channel connectivity policy describes the allowed connections between entities within a distributed system. This policy is an IEMP policy limiting which entities may directly communicate via channels provided by the PCS. The resource management policy describes how the shared communication resources used to implement channels are to be allocated between channels. The PCS provides the following functionalities:

- Management of shared communication resources to provide channel separation.
- Authentication of entities.
- Protection of data confidentiality.
- Verification of data integrity.

Using a PCS allows system designers to locate partitions on different processors without introducing new threats to data confidentiality or integrity due to inter-processor communication between those partitions[12]. It encrypts data and does entity authentication before allowing data to flow. The PCS extends MILS policies (data isolation, information flow, periods processing, damage limitation between partitions) to include end-to-end enforcement of policies. Although the PCS guarantees separation in the network, it does not have control over each partition, so guards and application security are still needed.

Wireless Network Example: The MILS wireless network utilizes devices (e.g., entity A, entity B) to send/receive messages, via the PCS, across a wireless network using an access point, and then to a wired network (e.g., the Internet). Wireless entity requests have to go through the access points before a decision is made to either grant or reject the request. Access points have two interfaces, one is a wireless interface that understands wireless protocols and another is a wired interface that allows entities to connect to the wired network using Ethernet. Security policies are implemented by access points before data is being transferred from wireless to wired networks. Wireless access points regularly transmit encrypted signals so the PCS components are aware of the existence of such access points and can use them for communication.

Figure 2 shows an example of a MILS wireless network implementation. Notice that some partitions may have an application without a middleware; this simplifies the design. Steps 1 through 8 show how a message gets transferred from entity A to entity B. Notice that the dashed line pattern indicates a wireless communication, whereas the solid line pattern indicates a wired communication. If any of the components along the message path rejects the message request, then an error will be generated and sent back to the requesting entity (entity A) informing it of request rejection.

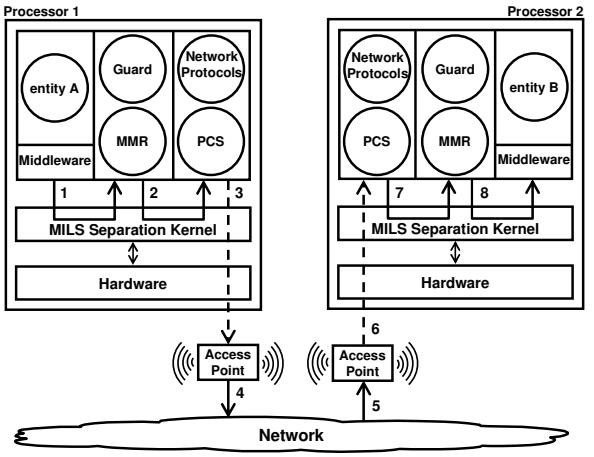

Fig. 2: MILS wireless network implementation.

- Step 1: The message that entity A sends is mediated by the CORBA middleware and if access is granted, then the message is sent to the MMR.
- Step 2: The MMR enforces finer granularity policies. If the message is of a specific type, then the MMR routes this message to an appropriate guard. If the guard determines that the content of the message is not in compliance with information flow policy, the guard will notify the MMR which will reject the message. If access is granted, then the message is sent to the PCS.
- Step 3: The PCS checks the network protocols and encrypts the message before it is transmitted over to a wireless access point.
- Step 4: The access point transmits the message over to the wired network.
- Step 5: The network sends the message to the destination access point.
- Step 6: The wireless access point transmits the message to the destination PCS.
- Step 7: The PCS checks the network protocols and decrypts the message and if access is granted, it will send the message to the MMR.
- Step 8: The MMR routes the message to an appropriate guard. If access is granted, then the

message is sent to the CORBA middleware that mediates the message before entity B receives it.

POLICY REFINEMENT

In our model, entities are divided into two subsets, trusted and untrusted. Trusted entities consist of trusted computing components (TCC) that are solely used to enforce the security policies of a given system such as the MMR, guard, and PCS. They take the transferred messages and security policies associated with the communication path as input and output the modified messages. The possible modifications of a message could be encryption, downgrade, or even deletion which means the information flow is illegal. All other entities within the system are untrusted. Applications, such as those that provide ordinary functionalities of a given system, fall into this category. Each untrusted entity is assigned a security classification. An untrusted entity is said to be single level if it processes data of one security classification and multi-level if it processes data of multiple security classifications.

Policy refinement is the decomposition of high-level policies into lower-level, more specific policies that can be enforced by the system[20]. In order to secure the system, security mechanisms need to be selected to enforce the desired policies. The enforcement procedure is implemented by incorporating security mechanisms which transform the application view of the system into its low-level implementation view. Transformation, in our system, is then defined in terms of the refinement steps for a security enforcement procedure of an application system, namely the procedure of plug-in TCC.

The goal of refining high-level policies into more specific policies is to reach an implementable configuration in which the information flow graph has no specific edge policies. The use of refinement will reduce the proof effort for secure components. In order to facilitate the transformation (or implementation) of a secure computer system, we present the following set of rules:

Transformation Rule 4: No direct communication channel is allowed between entities classified at different classifications.

Transformation Rule 5: Information flow between untrusted entities having different classifications must be mediated. More specifically, it must pass through a trusted computing component.

Transformation Rule 6: Typed messages must be sanitized by a guard with the same type. An entity is connected with a corresponding guard based on the message type it transfers.

Transformation Rule 7: Inter-partition communication must pass through an MMR.

Transformation Rule 8: Inter-processor communication must pass through a PCS.

Policy Conjunction Rule 9: Policy of a guard is the union of security policies associated with the communication channels that it mediates and IEMP.

Policy Conjunction Rule 10: Policy of an MMR is the union of inter-partition and edge policies and IEMP.

Policy Conjunction Rule 11: Policy of a PCS is the union of inter-processor policies and IEMP.

The principle operations of policy refinement are the following:

1. Allocate entities to different partitions according to their security classification to ensure data isolation. This is the application of Transformation Rule 4.
2. Integrate a guard to enforce policies associated with communication channels. This is the application of Transformation Rule 5. The operation can be further refined into three steps. First, connect each entity with a corresponding guard based on the message type it transfers, following Transformation Rule 6. Second, combine the policies of the original channels using IEMP to form the policy of the guard, according to Policy Conjunction Rule 9. Last, assign each communication channel a policy stipulating the type of messages it can transfer, which is the type of message the guard sanitized, following Transformation Rule 6.
3. Integrate an MMR to enforce inter-partition and edge policies. According to Transformation Rule 7, we further refine the above implementation by incorporating an MMR. Inter-partition communication must pass through and be mediated by an MMR. Allowed information will then be routed to a corresponding guard based on its type (e.g., a GIOP message is sent to a GIOP Guard). It takes two steps to accomplish the transformation. First, incorporate an MMR between a guard and an untrusted entity, following Transformation Rule 7. Second, form the policy of MMR, according to Policy Conjunction Rule 10.
4. Integrate a PCS to enforce inter-processor policies. According to Transformation Rule 8, we further refine the above implementation by incorporating a PCS. Inter-processor communication must pass

through and be mediated by a PCS. Allowed information will then be routed to a corresponding PCS, forming the policies of PCS, according to Policy Conjunction Rule 11.

Different policy models in the literature have been developed to restrict information access. Although most systems are restricted to a single policy model to provide security[15], our proposed approach is capable of dealing with multiple policies from different models that are being enforced by the system.

Manley et al.[11] pointed out a major security concern with wireless security: they stated that 70% of the data transmitted through wireless access points is unencrypted. They proposed a framework of a reliable wireless security policy and examined the wireless security policies of the Department of Defense based on their framework. Arbaugh[3] argued that wireless security requires slightly different thinking from wired security primarily because it gives potential attackers easy medium access which increases systems' security threat.

Although significant work in developing policy refinement strategies has been done, several researchers, including Bandara et al.[4], pointed out that more issues remain to be addressed. They presented an approach to policy refinement that allows inferencing of low-level actions that satisfy a high-level goal. Zhou and Alves-Foss[20] proposed architecture-based refinement techniques for the design of multi-level secure systems.

CONCLUSIONS

This paper outlines a wireless inter-enclave multi-policy technique that provides system security managers with a framework for supporting the enforcement of diverse security multi-policies in MILS, a high assurance computer system design for security and safety-critical multi-enclave systems.

In order to address new challenges of wireless security, system security managers should constantly review policies to ensure that new threats are covered by the existing policies. As security attacks and communication environments are constantly changing, security researchers should implement a thorough assessment in order to identify new security vulnerabilities. In order to minimize security risks, a better understanding of wireless technology and its effect on the enforcement of security policies is essential. The relationship between wireless technology and security engineering introduces new challenges that need to be investigated. The approach proposed in this paper is an important step towards defining (understanding) this relationship.

ACKNOWLEDGEMENTS

We wish to acknowledge the United States Air Force Research Laboratory (AFRL) and Defense Advanced Research Projects Agency (DARPA) for their support. This material is based on research sponsored by AFRL and DARPA under agreement number F30602-02-1-0178. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of AFRL, DARPA, or the U.S. Government. We also wish to acknowledge the anonymous reviewers and journal editors for reviewing this paper.

REFERENCES

1. Alves-Foss, J., W. S. Harrison, P. Oman, and C. Taylor, 2006. The MILS architecture for high assurance embedded systems. International Journal of Embedded Systems, 2 (3/4): 239-247.
2. Alves-Foss, J., C. Taylor, and P. Oman, 2004. A multi-layered approach to security in high assurance systems. In Proceedings of the 37th Annual Hawaii International Conference on System Sciences.
3. Arbaugh, W. A., 2003. Wireless security is different. Computer, 36 (8): 99-101.
4. Bandara, A. K., E. C. Lupu, J. Moffett, and A. Russo, 2004. A goal-based approach to policy refinement. In Proceedings of the 5th IEEE International Workshop on Policies for Distributed Systems and Networks, pp: 229-239.
5. Bell, D. E. and L. J. LaPadula, 1976. Secure computer systems: Unified exposition and MULTICS interpretation. Technical Report ESD-TR-75-306, MITRE Corporation MTR-2997 Rev. 1.
6. Department of Defense, 1985. Trusted computer system evaluation criteria. Computer Security Center. No. DoD 5200.28-STD.
7. Harrison, W. S., N. Hanebutte, P. Oman, and J. Alves-Foss, 2005. The MILS architecture for a secure global information grid. Crosstalk: The Journal of Defense Software Engineering, 18 (10): 20-24.

8. Heimdahl, M. P. E. and C. L. Heitmeyer, 1998. Formal methods for developing high assurance computer systems: Working group report. In Proceedings of the 2nd IEEE Workshop on Industrial Strength Formal Specification Techniques, pp: 60-64.
9. Lockheed-Martin, Boeing, Rockwell Collins, Green Hills Software, LynuxWorks, Objective Interface, and the University of Idaho, 2003. Protection profile for partitioning kernels in environments requiring augmented high robustness. Version 1.3, submitted for the Open Group and the Information Assurance Directorate of the National Security Agency.
10. Malaney, R. A., 2004. A location enabled wireless security system. In Proceedings of the IEEE Global Telecommunications Conference, volume 4, pp: 2196-2200.
11. Manley, M. E., C. A. McEntee, A. M. Molet, and J. S. Park, 2005. Wireless security policy development for sensitive organizations. In Proceedings of the IEEE Workshop on Information Assurance and Security, pp: 150-157.
12. Objective Interface Systems, 2005. Protection profile for partitioning communications systems in environments requiring high robustness. Draft V0.85.
13. Perrig, A., J. Stankovic, and D. Wagner, 2004. Security in wireless sensor networks. Communications of the ACM, 47 (6): 53-57.
14. Rushby, J. M., 1981. Design and verification of secure systems. In Proceedings of the 8th ACM Symposium on Operating System Principles, pp: 12-21.
15. Spencer, R., S. Smalley, P. Loscocco, M. Hibler, D. Andersen, and J. Lepreau, 1999. The Flask security architecture: System support for diverse security policies. In Proceedings of the 8th USENIX Security Symposium, pp: 123-139.
16. Verma, D., 2000. Policy-Based Networking: Architectures and Algorithms. New Riders Publishing.
17. Wahsheh, L. A. and J. Alves-Foss, 2006. Specifying and enforcing a multi-policy paradigm for high assurance multi-enclave systems. Journal of High Speed Networks, 15 (3): 315-327.
18. Wahsheh, L. A. and J. Alves-Foss, 2007. Using policy enforcement graphs in a separation-based high assurance architecture. In Proceedings of the IEEE International Conference on Information Reuse and Integration, pp: 183-189.
19. White, P., W. Vanfleet, and C. Dailey, 2000. High assurance architecture via separation kernel. Draft.
20. Zhou, J. and J. Alves-Foss, 2007. Security policy refinement and enforcement for the design of multi-level secure systems. Journal of Computer Security. In press.