

# A Note on Mapping User-Oriented Security Policies to Complex Mechanisms and Services<sup>1</sup>

Cynthia Irvine  
Naval Postgraduate School  
Monterey, CA

Tim Levin  
Anteon Corporation  
Monterey, CA

**Abstract.** *The quality of service framework in a heterogeneous computer network environment may provide users and applications with a wide range of security mechanisms and services. We propose a simplified user security interface and a method for mapping this interface to complex underlying security mechanisms and services. Additionally, we illustrate a mechanism for mapping multiple security policies to the same user security interface.*

## 1 Introduction

In a heterogeneous computer network<sup>2</sup>, the user can be presented with a wide range of security services and enforcement mechanisms [5] instituting security policies from various security domains. The security domains can be geographically diverse (e.g., subnets traversed to a remote internet destination) and layered (e.g., application security policies versus network security policies). The problem of mapping security mechanisms between different network layers is identified in the literature (e.g., see [3] [10] ), as is the composition of policies and policy domains (e.g., see [2] [6] [11] ). However, the problem remains as to how users and administrators can understand and easily interact with a wide range of security services and mechanisms. This note address the translation of a simplified user abstraction of security to detailed underlying mechanisms, such that users can be presented with a coherent user-level view of available security options.

## 2 User Security Interface

In the network computing context, users may request the execution of “tasks,” which are scheduled by an underlying control program (e.g., a Resource Management System, “RMS”) to execute on local or remote computing resources. The execution of a task may access a variety of network resources, such as: local I/O device bandwidth, internetwork bandwidth; local and remote CPU time; local, intermediate (e.g., routing buffers) and remote storage. Each resource may have its own security constraints. One cannot expect users or even application developers to understand the implications of the detailed interfaces of all of these mechanisms. Therefore a simplified, generalizable user-interface is called for.

We present a framework for mapping a simple user interface to an arbitrarily complex set of detailed security mechanisms. We will use the following simple user interface for illustration,

---

1. Funded through MSHN, a DARPA/QUORUM project.

2. A network comprising a variety of software and hardware implementations for processing, networking and storage.

however other simple taxonomies might suffice<sup>1</sup>. We envision a QoS-like interface in which the user may specify the degree or “level” of security service, in general, that is to be applied to the processing of the network task. Such a level might be as simple as:

`user_security_level ::= [high | medium | low]`

Thus, a user QoS request might appear like this:

`QoS Request ::= task_specifier, user_security_level, performance_vector, other_factors`

## 2.1 Application and System Security

Various quality of service approaches are including security as one of the vectors of service provided to the user [4] [9] [12] [14]. It is apparent that, if a QoS system is going to provide choices to the user with respect to security, the underlying mechanisms need to provide variable security, and that the network security policy(s) need to allow security to vary.

However, computer security has been envisioned traditionally at the *system* level [1] [2]. Users and applications were constrained by underlying security mechanisms to behave in ways that conformed to the *system* security policy, and system security policies did not allow the security requirements to vary. For present-day network security, considering the network and the OS(s) to be the “system,” there has been some shift of emphasis from *system* security to *application* security. That is, each application (e.g., an email program) may present a security environment to its users, and is responsible for protecting the user’s rights and data in that environment and in the network. We believe that the needs for application-level security must be accommodated; however, network system security policies cannot be ignored in the process, rather, different levels of policy must work in harmony. Thus, given that the over-arching network *system* security policy demands some minimum degree of system security policy enforcement, application-level selections for quality of security service may be provided to users to any degree of security over and above those system minimums. That is to say, an application can always provide more security, than the minimum required by the base system security policy, while still complying with that policy. Similarly, application enforcement of user security *maximums* might be possible, e.g., to limit processing expenditures, if those maximums are within the bounds of the underlying security policy(s).

We refer to services and mechanisms that allow a range of security behaviors as “variant.” Variant security mechanisms may be used within a resource management context, for example, to effect adaption to varying network conditions. Also, if the policy mechanism is variant, the control program may offer quality of security service choices to the users and their network tasks.

## 2.2 Security Terminology

Before discussing the mapping of a simplified user security interface to complex underlying mechanisms, some security mechanism terminology is presented (see [5] for further explanation).

Users and applications on the network are presented with various security *services* (e.g., data-flow confidentiality, non-repudiation). A security service may be used to implement one or more secu-

---

1. TCSEC evaluation classes or Common Criteria profiles could be used

curity *policies* (organizational or automated [15] ), which are in turn implemented by one or more security *mechanisms*. As described above, some mechanisms provide fixed services, and some are *variant*<sup>1</sup>. Additionally, the RMS may make choices for the user regarding variant security mechanisms, as part of its schedule formulation or adaptive re-scheduling.

Each security mechanism is associated with a *service area*, which indicates the general topographical component of the network in which the security or protection is effective. We identify three service areas: end system (e.g., a client or server system), intermediate node (e.g., routers, switches), and network connection (i.e., the “wire” connecting various systems and nodes). Security mechanisms associated with end systems and intermediate nodes protect resources (e.g., data and programs) that are associated with a node or system; for network connections, we are concerned with mechanisms for protecting information that is physically in transit.

## 2.3 Mapping User Security Interface

The elements of the simple user interface are mapped to detailed mechanism invocations via a *translation matrix*. Table 1 shows a mapping of our example user security scale (viz, low,

**Table 1: Example User Security Translation Matrix**

Security Service	Service Area	User Security Scale		
		Low	Medium	High
Data Confidentiality	ES	none	OS access controls	B3-level DAC
Data Integrity	Wire	none	DES 56-bit key	DES 128-bit key
Login Authenticity	ES	OS I & A	B1-level I & A	Public key certificates
Message Nonrepudiation	ES	none	OS auditing	Digital Notary Service

---

1. Variant mechanisms offer the user various “degrees,” or strengths, of security (viz., over and above some minimum requirement).

**Table 2: Security Translation Matrix with Network Modes**

			User Security Scale		
Security Service	Service Area	Network Mode	Low	Medium	High
Data Confidentiality	ES	normal	none	OS access controls	B3-level DAC
		impacted	none	OS access controls	OS access controls
		under attack	OS access controls	B3-level DAC	B3-level DAC
Data Integrity	Wire	normal	none	DES 56-bit key	DES 128-bit key
		impacted	none	none	DES 56-bit key
		under attack	DES 56-bit key	DES 56-bit key	DES 128-bit key
Login Authenticity	ES	normal	none	B1-level I & A	Public key certificates
		impacted	none	B1-level I & A	OS-level I & A
		attack	OS I & A	B1-level I & A	Public key certificates
Message Nonrepudiation	ES	normal	none	OS auditing	Digital Notary Service
		impacted	none	none	OS-level auditing
		under attack	OS auditing	Digital Notary Service	Digital Notary Service

medium, high) to a heterogeneous network which has several *variant* security services. Each level in the user security scale is characterized by one or more mechanisms for each security service. Also, a particular security mechanism may be mapped to more than one user security level, e.g., in Table 2, 56-bit keyed DES is the mechanism to satisfy data integrity services in the network attack mode for both low and medium user security levels.

In this example, the network has end systems with both simple OS discretionary access control (DAC) and with class B3 evaluated [1] DAC, indicating that the system policy allows OS-level DAC to be enforced with a range of mechanisms. There are also a variety of integrity, authenticity and nonrepudiation services available. With this mapping, the user is not offered all combinations of variant services; instead, the security administrator or system security engineer has pre-selected various specific mechanisms and settings that map to the three choices offered to the user. The

idea here is to provide the user/task with a virtual network in which all elements possess consistent assurance qualities, e.g., effectiveness and/or worthiness. Thus, the network security architecture is coherent with respect to each of the service requirements and has no weak link. These example mappings illustrate mechanisms to govern users at the *system* level; mapping pre-selections could be made also at the *application* level, but that is not illustrated here.

This type of translation matrix can be used to both: (1) translate abstract levels or scales of security services to specific settings in the underlying mechanisms (as is illustrated above), and (2) given a set of security mechanisms (e.g., from a distributed system), derive the abstract level of service that is available (e.g., the greatest lower bound).

Thus, users can indicate the desired security *degree* or “*level*” of their connection, perhaps as part of a QoS request (see Section 2 on page 1). The underlying RMS or control program would be responsible for assigning security services and resources to the user that would meet the security profile indicated by the translation matrix. If corresponding services or resources could not be found to meet the user request, then the RMS would need to negotiate different degrees of service with the user, or perhaps use a default translation.

## 2.4 System Architecture

The translation matrix can be implemented in a variety of ways. For example, a globally-accessible directory could be managed by a security policy server, and be accessed by the RMS as needed to translate user requests. Alternatively, the matrix could be implemented as an RMS internal table, and managed by an RMS administrative tool.

## 2.5 Alternative Frameworks

As an alternative to the highly abstract user interface described here, detailed numeric measurements can be applied to each mechanism. Novell defines a security taxonomy within its crypto environment [7], with numeric security-strength indicators for the various components. Wang and Wulf [16] have organized a security taxonomy in a hierarchical fashion to provide a detailed metric for security services. Such a system could present users with a numbering system with which to indicate the desired strength of each security mechanism, and present summary numbers to indicate the overall strength of certain subsystems or sub-networks. However, we feel that much work needs to be done to standardize such metrics, and to educate users as to their meaning.

## 3 Dynamic Security Policy Support

With a *dynamic* network security policy [9], the security restrictions and available security services depend on the network status or “mode” (e.g., normal, impacted, emergency, etc.) [8].

Access to a predefined set of alternate security policies allows their functional requirements and implementation mechanisms to be examined with respect to the overall policy prior to being fielded, rather than depending on an ad hoc review. For example, during an emergency, a military commander might decide to forgo certain security protocols in order to get some important information transmitted quickly. This decision changes the security policy, but the actual policy arrived at may not be clearly understood.

If dynamic policies are created before deployment of the computer network, the network can

respond to changing environments, while avoiding the confusion of ad hoc changes. A corporate intranet might have a mode indicating that the system is under attack from the internet. In this mode, it might be desired for a higher degree of network security to be in place. A military network might have an “emergency” mode indicating that there is a physical threat to the facility, and that command messages (only) which would normally be encrypted and signed, need to go out with the highest bandwidth available, disregarding cryptographic security. An ISP might have an “impacted” mode in which certain optional user security services would be curtailed for efficiency. In each of these cases, the changes to the security mechanisms would be predefined and limited to meet the desired alternate security policy.

In Table 2, some hypothetical network modes are included in the translation matrix from Table 1, showing how the “user security level” mappings would change, per mode. The modes are: normal, impacted and attack, as described above.

## 4 Conclusion

A security translation matrix can be used to provide users with a simplified representation of application and system security. Such a matrix can be used to translate user interfaces to a wide range of mechanisms, independent of how the mechanisms are related or distributed in the network. This mechanism can be used to support both variant and dynamic security policies.

## References

- [1] --, Department of Defense Trusted Computer System Evaluation Criteria, Department of Defense, DoD 5200.28-STD, December 1985
- [2] --, Trusted Network Interpretation of the Trusted Computer System Evaluation Criteria, National Computer Security Center, NCSC-TG-005, Version-1, July 1987
- [3] Aurrecoechea, C., Campbell, A., and Hauw, L. “A Survey of Quality of Service Architectures”, *Multimedia Systems Journal*, Special Issue on QoS Architectures, 1996.
- [4] Hensgen, D., Kidd, T., St. John, D., Schnaidt, M.C., Siegel, H. J., Braun, T. Maheswaran, M., Ali, S., Kim, J-K., Irvine, C. E., Levin, T., Freund, R., Kussow, M., Godfrey, M., Duman, A., Carff, P., Kidd, S., Prasanna, V. Bhat, P., and Alhusaini, A., “An Overview of the Management System for Heterogeneous Networks (MSHN),” *Proceedings of the 8th Workshop on Heterogeneous Computing Systems (HCW '99)*, San Juan, Puerto Rico, Apr. 1999, pp 184-198.
- [5] Irvine, C., and Levin, T., *Toward a Taxonomy and Costing Method for Security Metrics*, NPS Technical Report, Forthcoming
- [6] Johnson, Dale and Thayer, E. Javier, Security and the composition of machines, In *Proceedings of the Computer Security Foundations Workshop*, pages 72-89, IEEE Computer Society Press, June 1988.
- [7] Juneman, R. R., *Novel Certificate Extension Attributes--Novel Security Attributes: Tutorial and Detailed Design*. Version 0.998, Novell, Inc. 122 East 1700 St., Provo, UT, August 1997.
- [8] Kim, Jong-Kook, Hensgen, D., Kidd, T., Siegel, H.J., St.John, D., Irvine, C., Levin, T.,

- Prasanna, V., and Freund, R., Priorities, Deadlines, Versions, and Security in a Performance Measure Framework for Distributed Heterogeneous Networks, Technical Report, Forthcoming.
- [9] Levin, T., and Irvine C., Quality of Security Service in a Resource Management System Benefit Function, NPS Technical Report, Forthcoming
  - [10] Nahrstedt, K., and Steinmetz, R., Resource management in networked multimedia systems. *IEEE Computer*, 28(5):52-65, May, 1995.
  - [11] Rushby, John, Composing trustworthy systems: A position paper, In Proceedings of the NATO RSG2 Working Conference on Composability, October 1991
  - [12] Sabata, B., Chatterjee, S., Davis, M., Sydir, J., Lawrence, T. "Taxonomy for QoS Specifications," Proceedings the Third International Workshop on Object-Oriented Real-Time Dependable Systems (WORDS'97), February 5-7, 1997, Newport Beach, Ca., pages 100-107
  - [13] Schantz, R. E. "Quality of Service," to be published in "Encyclopedia o Distributed Computing," 1998.
  - [14] Schneck, P. A., and Schwan, K., "Dynamic Authentication for High-Performance Networked Applications," Georgia Institute of Technology College of Computing Technical Report, GIT-CC-98-08, 1998.
  - [15] Stern, D. F., On the Buzzword "Security Policy", Proceedings of 1991 IEEE Symposium on Security and Privacy, 1991, Oakland, Ca., pages 219-230.
  - [16] Wang, C. and Wulf, W. A, "A Framework for Security Measurement." Proceedings of the National Information Systems Security Conference, Baltimore, MD, pp. 522-533, Oct. 1997.

