

Discussion Topic: The Security Contract – An Evolving Definition

Proposer: Edward A. Feustel
Institute for Security Technology Studies
Dartmouth College
Hanover, NH 03755
efeustel@ists.dartmouth.edu
(603)646-0671

With the aid of: Terry Mayfield
Institute for Defense Analyses
Alexandria, VA 22311-1772
mayfield@ida.org

Information security undergoes constant change in focus and scope as new technologies emerge, new uses are identified, and new system designs evolve. Early in the security of computing, efforts were focused on integrity (preventing unauthorized modification in a shared computing environment), with type-based languages, virtual machines, monitors, checksums, two-stage commits, etc. Efforts were also focused on security, but mainly in the procedural and physical access in computer centers. Later, the issue of safe sharing of sensitive information became a paramount issue. Today, with distributed computing and large-scale networking, efforts are being focused on qualities of availability (prevention of denial of service), accountability, and assurance (quality of implementation), while also needing to retain focus on confidentiality and integrity issues. This widened focus has made the scope of information security increasingly complex. The parameters that are needed for specifying and enforcing security policies in large-scale networked systems must be captured and used in an interoperable fashion to execute the “security contract” in various portions of the system. Clearly aspects previously seen as factors in quality of service (QoS) contracts are now being claimed by the security contract under which users of shared resources operate.

This session will focus on the needs for this security contract or set of contracts and the policies they describe for each interaction they cover. What are the forms of these contracts and what factors determine their form? What parameters are involved in a security contract? How are security contracts used in different layers of a distributed system? How can intentional naming systems and discovery protocols or publish and subscribe protocols assist in negotiating a security contract between two heterogeneous systems. How will the security contract play in active networks? These and more questions are emerging as we begin to see policy-driven systems become more central to distributed systems in large-scale networks.

The security contract includes factors that determine:

- whether an authorization to perform an action (sequence of actions, graph of actions) is granted;
- how that authorization is used to complete the action; and
- whether obligations must be satisfied prior to, during, or on completion of an action.

The security contract usually references resources, e.g., processes and data that are shared, explicitly and implicitly.

Previously we were concerned primarily with enforcing the security contract on a single computing platform and upon a single simultaneous action. Now we attempt to enforce it on *automations of user-specified processes*, where computational tasks assume the roles of requestor of services and provider of services, and on communications between these processes. In the future we will wish to enforce a security contract during interactions depicted by graphs whose nodes are tasks and whose arcs denote either service requests or service responses where tasks may execute concurrently. This enforcement may occur in a heterogeneous environment consisting of different platforms and/or operating systems, in different security technology domains, and in different security management domains.

Previously our model for enforcement of the security contract focused on the *subject-object* matrix and “access rights”. This model may not handle the range of contracts desired in the case of distributed systems: our model may have to focus much more on the security context of any particular computational task (principal(s), actor(s), collaborative task(s), current and/or elapsed time, interface and operation to be used, implementation(s), degree of trust in any provided parameter, and other information regarding the security state and its history). In the case that we have a requesting service, a responding service, and an object to be acted upon, we may need to consider *factors* relating to the requestor task, the responder task, the object of the action and the histories of all three to determine whether the action is to be authorized and under what condition(s). We may also wish to consider whether the collection of factors is to be static or dynamic as well as whether the values associated with the factors are mutable or immutable and in what time scale.

The goal of this session is to address the following questions:

1. During the next five to seven years, what factors should be included in a security contract? Why?
2. Which factors, though desired for personal, business, or governmental reasons, should not be included? Why?
3. If it is likely that the number of factors required will increase, how may we design implementations so that their addition will be evolutionary in character rather than revolutionary in character?

4. Is it possible to evolve easily from current contracts to more sophisticated ones?
5. What must be standardized now/soon to insure the greatest possibility of interoperability across technology and management domains while enforcing the security contract?
6. What emerging methods can support new thinking on security contracts (e.g., Discovery Protocols, Intentional Naming, Micro-protocol toolkits, and Agents)
7. To what should a security contract apply, e.g., interaction, transaction, principal, platform, or management domain?

References:

Adjie-Winoto, William, et al, "The Design and Implementation of an Intentional Naming System," 17th ACM Symposium on Operating Systems Principles (SOSP '99), Published in *ACM Operating Systems Review*, 34(5):186-201, Dec 1999.

Pascoe, Robert, "Building Networks on the Fly," *IEEE Spectrum*, pp 61-65, March 2001.

MIT Laboratory of Computer Sciences, *Project Oxygen, Project WIND, and SFS*;
<http://www.oxygen.lcs.mit.edu/network.html>

A.Brady Montz and Larry Peterson. Controlled Flexibility in System Design.
<http://www.cs.princeton.edu/nsg/scout/papers/sigops98.ps>

B. Lampson, M. Abadi, M. Burrows, and E. Wobber. Authentication in distributed systems: Theory and practice. *ACM Trans. Computer Systems* **10**, 4 (Nov. 1992), pp 265-310.

Roshan Thomas and Ravi S. Sandhu. Conceptual foundations for a model of task-based authorizations. In *IEEE Computer Security Foundations Workshop 7*, pages 66--79, Franconia, NH, June 1994.

D. Tennenhouse, J. Smith, W. Sincoskie, D. Wetherall, and G. Minden. A Survey of Active Network Research. *IEEE Communications Magazine*, pages 80--86, January 1997.

Edward A. Feustel and Terry Mayfield. *Unmet Information Security Challenges for Operating System Designers*. *Operating Systems Review*, 32(1):3--22, January 1998.