Toward A Unified View of Intrusion Detection and Security Policy

Matt Blaze
AT&T Labs - Research, mab@research.att.com

Angelos D. Keromytis
Columbia University, angelos@cs.columbia.edu

Sal Stolfo
Columbia University, sal@cs.columbia.edu

7 April 2002

1 Proactive vs. Reactive Security

1.1 Policy Enforcement - "Proactive" Security

Security policy enforcement mechanisms, such as access control, trust management systems [2], and higher-level policy languages [1] focus on *preventing* unauthorized behavior. In general, such mechanisms focus, directly or indirectly, on the configuration of "policy enforcement points," which determine whether proposed actions are permissible according to policy; this model was best formalized in the trust management literature, but applies across the security policy spectrum.

Much recent research (and a few practical systems) aim proactive security mechanisms that are "policy driven," in the sense that a security policy specification is used to generate an implementation that guarantees that the resulting system has the security properties specified. This work, however interesting, is not the subject of this position paper.

Another interesting area of current work is concerned with providing high level abstractions at multiple enforcement points across multiple layers of a system. Although this work is also important, it, too, is not the subject of this position paper.

1.2 Intrusion Detection – "Reactive" Security

Although there is still quite a bit to do in the area of proactive security, the policy enforcement model is not always sufficient in practice to capture all aspects of system security. In particular, the policy enforcement model implicitly forces specification of those aspects of policy that can be enforced at an enforcement point.

Not all aspects of what we intuitively consider security policy are readily expressible under the proactive enforcement model. For example, (high-level) security policies in real organizations are often concerned with "after the fact" controls such as audit, traceability, logging, and incident response – they are concerned with *detecting* unauthorized behavior. These "reactive" aspects of policy are typically implemented through intrusion detection systems and similar mechanisms. Here the focus is not on preventing violations, but on effectively detecting and responding to them.

Reactive security is rather different in character from proactive security. While, on the surface, the prophylactic nature of proactive security would seem always to be preferable to even the most diligent reactive mechanisms, in practice things are not so simple. In particular, it is not always possible to specify a complete security policy strictly in terms of proactive controls at policy enforcement points. The mechanisms and controls that implement reactive aspects of policy can often be much more loosely coupled to the systems that they control than can policy enforcement points (which are typically implemented at the lowest layers of a system) can. IDS systems, for example, can take advantage of data from a wide range of system components, and thus can sometimes reveal and respond to global behavior patters that are not visible to the lower-level, if proactive, policy enforcement points.

But intrusion detection, however rewarding it may be, is not in and of itself the subject of this position paper.

2 Unifying Proactive and Reactive Policies

An unfortunate consequence of the breadth and complexity of "security policy" as it exists in modern large-scale distributed systems is that even if the proactive aspects of policy can be specified and managed in a single place, the reactive aspects of the policy must be specified, managed and implemented in another.

Some aspects of policy fit clearly in to either a proactive or a reactive model. For example, contrast the following two examples:

- Access control. Filesystem accesses are typically governed by a proactive access control policy that is always enforced as access are requested and that should prevent any unauthorized access from ever occurring. This implies that, for each request on a file, a lookup is made on the access control matrix to determine the privileges of the user/process relative

to the file. In the various Unix flavors, the matrix is implemented in a distributed manner, by associating permission bits and owner/group information with filesystem objects. Because the determination can be made very quickly, security checks are always performed.

- Audit. On the other hand, security against patterns of fraud in bank, securities, or credit card transactions is often primarily based on analysis of audit trails. A reactive security policy (concerned with fraud detection and consistency) is evaluated on the transaction log, and any suspect behavior is flagged, with the aim of detecting (and thereby discouraging) wrongdoing, without imposing undue costs on individual transactions.

Although it may be clear here that the "file access control" policy is proactive while the "bank audit" policy is reactive, this may not be at all clear to the administrator responsible for creating and managing the policy in a real system. Worse, separating the mechanisms for specifying these policies makes it difficult to employ tools that might look at overall security policy and that might consider interactions between different policy elements.

Indeed, which aspects of policy are best implemented through proactive mechanisms and which aspects through reactive mechanisms may depend on implementation details that are hidden from the policy writer (and that are subject to change over time or based on particular conditions). For example, a network security policy may be implemented as a mix of intrusion detection and access control policies: if the traffic volume is high, simple access control rules may be enforced by the firewall, permitting authorized traffic through. If the firewall if not heavily utilized (or if the appropriate software/functionality is present), an IDS policy may be applied on the data streams with the hope of detecting more complex patterns of unauthorized behavior. Convention proactive and reactive policy specification tools do not allow for this possibility, and make such systems quite difficult to analyze.

We suggest that a higher-level abstraction is in order here, one that hides low-level details of how a policy is implemented whether proactively or reactively. This opens a potentially rich area of research in high-level policy specification tools, analysis mechanisms, and implementation architecture.

We welcome comments and suggestions.

References

[1] A. D. Keromytis. *STRONGMAN: A Scalable Solution To Trust Management In Networks.* PhD thesis, University of Pennsylvania, Philadelphia, December 2001.

[2] M. Blaze, J. Ioannidis, and A.D. Keromytis. Trust Management for IPsec. In *Proc. of Network and Distributed System Security Symposium (NDSS)*, pages 139–151, February 2001.