Policy Evolution: Autonomic Environmental Security

Patrick McDaniel
Systems and Internet Infrastructure Security (SIIS) Laboratory
Pennsylvania State University
mcdaniel@cse.psu.edu

Abstract

Security policy in contemporary computing systems is largely inert. For this reason, reaction to changes in an environment requires manual (administrator) intervention. Futhermore, because policy does not reflect instantaneous conditions, it cannot comprehensively address emergent threats. This work seeks to investigate reactive policy through a theory and practice of *policy evolution*. Policy evolution transforms existing security configuration and authorization policy in response to observed conditions. This investigation applies formal modeling to the problem of autonomic security. The expected contribution is a general theory and linguistic elements for automated policy evolution.

1 Problem Statement

Security policy is the primary tool by which we control access to sensitive artifacts and infrastructure. Contemporary policy systems allow administrators to easily formulate and enforce enterprise-wide governing principles over these artifacts. However, policy is fundamentally inert: changes to the security posture of the environment are only reflected in system behavior after manual modification of policy elements. Because of a near universal lack of management resources, such changes to policy tend to only occur in practice when they are mandated by pressing needs. Hence, the validity of the guiding principals and underlying policy tends to decay over time. Moreover, instantaneous shifts in the hostility or reliability of an environment can and will led to unusable and vulnerable systems [McD03b].

Falling under the rubric of *autonomic computing*, recent efforts have sought to address the increasingly complex configuration and maintenance requirements of modern operating environments through automated system self-regulation [Coh03, KC03]. Such solutions are much more than merely adaptive systems, but seek to automatically reconfigure, optimize, heal, and protect the environment in response to changing conditions.[1] It is this last aspect that bears special attention: understanding how and when to alter protection mechanism behavior has deep implications to the security of an environment. An overly aggressive mechanism can lead to inefficiencies, and an overly conservative mechanism can lead to vulnerability. This work will explore how autonomic protection mechanisms can be implemented through *policy evolution*.

Policy evolution is the automated change in the behavior or structure of policy as directed by observed characteristics of the environment. Systems evolve policy for the same reasons that administrators change them: to reflect new security requirements, to make systems more responsive, or to allow for more efficient enforcement. Evolution allows the system not only to recognize when a policy change is necessary, but also to decide what and how to change it. To illustrate, consider a "use it or lose it " policy evolution heuristic. A security system operating under this heuristic evolves by removing rights that are not exercised (e.g., assigned but never used). Hence, evolution would converge on a usage-based *least privilege* policy. Of course, depending on the content, it may be advantageous to less aggressively evolve or exempt some policy from evolution, e.g., administrator access. Note that this strategy is a generalization of commonly used in short-lived credentials, e.g., in credentials revalidation strategies of SKPI/SDSI [RL96, EFL$^+$99].

How policy evolves over time depends on the goal of the particular evolutionary goal. In the preceding example, the goal of the system was to reduce over-specified rights. Other systems may attempt to optimize enforcement, to reduce maintenance by anticipating future needs, to mitigate rights abuse, or to address some new operational constraint or threat. The chief challenge of this proposed work is to understand how policy behavior and indirectly the quality and security of a computing environment can be enhanced by policy evolution.

[1] The secure assessment of environmental conditions is an immensely complicated and difficult task. We defer these largely systemic issues to our prior work on the subject [McD03a].

2 Background and Prior Research

Policy has been used in different contexts as a vehicle for representing authorization and access control [WL93, BFL96, CC97, WL98, RN00, LGF03], peer session security [ZSC+00], quality of service guarantees [BH99], and network configuration [Bel99, BMNW99]. These approaches define a policy language or schema appropriate for their target problem domain. The problem of automated policy synthesis has only been tangentially addressed, and evolution not at all. In the two-party case, the emerging Security Policy System (SPS) [ZSC+00] defines a framework for the specification and reconciliation of security policies for the IPSec protocol suite [KA98]. Policy negotiation is largely limited to the intersection of specified data structures. In the multi-party case, the DCCM system [DBH+00] provides a negotiation protocol for provisioning. DCCM defines a session policy from the intersection of policy proposals presented by each potential member. Each proposal defines a range of acceptable values along a multi-dimensional policy structure. Hence, reconciliation in these systems is largely based on the intersection of policy schema. In prior work, we developed a general model for provisional policy. We further demonstrated the fundamental intractability of *policy composition* [McD01, WJML04], the construction of a single unifying policy from collections of possibly conflicting policies, under this model.

Language-based approaches for specifying authorization and access control have long been studied [WL93, CC97, WL98, RN00], but they generally lack any semantics or mechanism for evolution. These systems typically identify rigorous semantics for the evaluation of authorization statements [TL04]. The PolicyMaker [BFL96] and KeyNote [BFIK99] trust management systems provide a powerful framework for the evaluation of credentials. Trust management approaches focus on the establishment of chains of conditional delegation defined in authenticated policy assertions. Hence, policy is dictated by entities to which session authority is delegated, rather than through the evoilution of environmental requirements [LWM03].

3 Goals and Objectives

Three questions immediately arise from the definition of policy evolution, a) what data do I collect to inform evolution, b) how do I extract meaningful information from the collected data and, c) what and how do I alter policy in response to learned facts? Answers to these questions will constitute the three main areas of investigation in this work: introspection, synthesis, and reconstruction.[2]

Data collection is performed through system *introspection*. There are many points for potential data collection: the inputs to the the policy evaluation, e.g., access requests, the specific enforcement acts e.g., granted and exercise rights, or measure the effect of policy on the system, e.g., performance. We expect that the set of appropriate measurements is again dependent on the evolutionary goal, and systems will tailor the set of measurement techniques to assess the quality of a policy. *Synthesis* gleans policy relevant facts from the data collection process. All the tools of machine learning are at our disposal here. We will use inference, extrapolation, categorization, or statistical analysis to determine where the policy is useful and where it can change. Policy *reconstruction* uses the identified facts to evolve the policy through planning or optimization. As a first step, new policies will be suggested to administrators. Evolved policies will be studied to axiomize good and bad evolution strategies, and characterize exceptional cases.

A second, albeit enormously important, phase of this work is to develop a mechanism and theory for the security of evolution: that is, how do I transition from one security posture to another without introducing further vulnerability? Moving from one policy to another requires that the current posture be reassessed, possibly leading to rights revocation or other transitional mechanisms. This investigation is explicitly outside the scope of this initial exploration, but will be revisited in immediate future work.

4 Research Plan

This work will extend our prior research in policy management [McD01, MP02] to evolution. Using the general model and theory of policy management, we will reason about the algorithmic complexity and structures of policy evolution. Policy linguistics supporting evolution will be explored, and mechanisms for their use in operational system will be deployed. This proposed work will investigate policy evolution systems in single host and distributed environments, and will focus solely on authorization policy.

Milestones

- Select candidate applications and services for modeling policy evolution.

[2] These evolutionary phases map loosely onto Kephart and Chess's notions of monitoring, analyzing, and planning [KC03]

- Identify evolutionary goals and requirements for single and multi-control systems.
- Develop linguistic constructs for policy evolution.
- Analyze algorithms for automatic policy evolution.
- Experiment with extensions and explore semantic scope of their use.

Deliverables

- Technical reports
- Analysis of policy evolution in sample applications
- Software/algorithms for automated policy evolution

Staffing and Budget

Principal Investigator: Patrick McDaniel
Graduate Research Assistant: Kevin Butler (tentative)

Budget attached.

References

[Bel99] S. Bellovin. Distributed Firewalls. *;login:*, pages 39–47, 1999.

[BFIK99] M. Blaze, J. Feignbaum, J. Ioannidis, and A. Keromytis. The KeyNote Trust Management System - Version 2. *Internet Engineering Task Force*, September 1999. RFC 2704.

[BFL96] M. Blaze, J. Feigenbaum, and Jack Lacy. Decentralized Trust Management. In *Proceedings of the 1996 IEEE Symposium on Security and Privacy*, pages 164–173, November 1996. Los Alamitos.

[BH99] David C. Blight and Takeo Hamada. Policy-Based Networking Architecture for QoS Interworking in IP Management. In *Proceedings of Integrated network management VI, Distributed Management for the Networked Millennium*, pages 811–826. IEEE, 1999.

[BMNW99] Yair Bartal, Alain J. Mayer, Kobbi Nissim, and Avishai Wool. Firmato: A novel firewall management toolkit. In *IEEE Symposium on Security and Privacy*, pages 17–31, 1999.

[CC97] L. Cholvy and F. Cuppens. Analyzing Consistancy of Security Policies. In *1997 IEEE Symposium on Security and Privacy*, pages 103–112. IEEE, May 1997. Oakland, CA.

[Coh03] David L. Cohn. Autonomic computing. In *Proceedings of the The Sixth International Symposium on Autonomous Decentralized Systems (ISADS'03)*, page 5. IEEE Computer Society, 2003.

[DBH+00] P. Dinsmore, D. Balenson, M. Heyman, P. Kruus, C Scace, and A. Sherman. Policy-Based Security Management for Large Dynamic Groups: A Overview of the DCCM Project. In *Proceedings of DARPA Information Survivability Conference and Exposition (DISCEX '00)*, pages 64–73. DARPA, January 2000. Hilton Head, S.C.

[EFL+99] C. Ellison, B. Frantz, B. Lampson, R. Rivest, B. Thomas, and T. Ylonen. SPKI Certificate Theory. *Internet Engineering Task Force*, September 1999. RFC 2693.

[KA98] S. Kent and R. Atkinson. Security Architecture for the Internet Protocol. *Internet Engineering Task Force*, November 1998. RFC 2401.

[KC03] J.O. Kephart and D.M. Chess. The Vision of Autonomic Computing. *IEEE Computer*, 36(1), 2003.

[LGF03] Ninghui Li, Benjamin N. Grosof, and Joan Feigenbaum. Delegation Logic: A Logic-based Approach to Distributed Authorization. *ACM Transactions on Information and System Security (TISSEC)*, 6(1), February 2003. RFC 2693.

[LWM03] Ninghui Li, William H. Winsborough, and John C. Mitchell. Beyond Proof-of-compliance: Safety and Availability Analysis in Trust Management. In *2003 IEEE Symposium on Security and Privacy*. IEEE, May 2003. Oakland, CA.

[LYR02] Jun Li, Mark Yarvis, and Peter L. Reiher. Securing Distributed Adaptation. *Computer Networks*, 38(3):347–371, 2002.

[McD01] P. McDaniel. *Policy Management in Secure Group Communication*. PhD thesis, University of Michigan, Ann Arbor, MI, August 2001.

[McD03a] P. McDaniel. On Context in Authorization Policy. In *8th ACM Symposium on Access Control Models and Technologies (SACMAT)*, pages 80–89. ACM, June 2003. Como, Italy.

[McD03b] P. McDaniel. Policy. In Hossein Bidgoli, editor, *Encyclopedia of Information Security*. Kluwer, 2003.

[MP02] P. McDaniel and A. Prakash. Methods and Limitations of Security Policy Reconciliation. In *2002 IEEE Symposium on Security and Privacy*, pages 73–87. IEEE, MAY 2002. Oakland, CA.

[RL96] R. Rivest and B. Lampson. SDSI A Simple Distributed Security Infrastructure. http://theory.lcs.mit.edu/ rivest/sdsi11.html, October 1996.

[RN00] T. Ryutov and C. Neuman. Representation and Evaluation of Security Policies for Distributed System Services. In *Proceedings of DARPA Information Survivability Conference and Exposition*, pages 172–183, Hilton Head, South Carolina, January 2000. DARPA.

[TL04] Mahesh V. Tripunitara and Ninghui Li. Comparing the Expressive Power of Access Control Models. In *Proceedings of 10th ACM Conference on Computer and Communications Security*. ACM, October 2004. Washington, DC.

[WJML04] H.B. Wang, S. Jha, P. McDaniel, and M. Livny. Security policy reconciliation in distributed computing environments. In *Proceedings of 5th International Workshop on Policies for Distributed Systems and Networks (Policy 2004)*. IEEE, June 2004. Yorktown Heights, NY, *to appear*.

[WL93] T. Woo and S. Lam. Authorization in Distributed Systems; A New Approach. *Journal of Computer Security*, 2(2-3):107–136, 1993.

[WL98] T. Woo and S. Lam. Designing a Distributed Authorization Service. In *Proceedings of INFOCOM '98*, San Francisco, March 1998. IEEE.

[ZSC+00] J. Zao, L. Sanchez, M. Condell, C. Lynn, M. Fredette, P. Helinek, P. Krishnan, A. Jackson, D. Mankins, M. Shepard, and S. Kent. Domain Based Internet Security Policy Management. In *Proceedings of DARPA Information Survuvability Conference and Exposition*, pages 41–53. DARPA, January 2000.