To be published in the ACM SIGSAC Review, 1992

METAPOLICIES I

A Position Paper
Presented at the ACM SIGSAC Special Workshop on
Data Management Security and Privacy Standards
San Antonio, Texas
December 3, 1991

by

Hilary H. Hosmer
Data Security Inc.
58 Wilson Road
Bedford, MA 01730

ACKNOWLEDGEMENT

This work was produced under a Small Business Innovative Research grant under the sponsorship of the Electronic Systems Division of the Air Force Systems Command, Hanscom Air Force Base, Bedford, MA.

ABSTRACT

Metapolicies, or "policies about policies", may become a powerful concept for developing the large, complex, and interrelated trusted systems that military, commercial and non-profit organizations need today. Metapolicies provide a framework for clarifying policies and for successfully coordinating security policies and subpolicies.

When there is only one security policy, metapolicies tend to be implicit, embedded, and fixed. When more than one security policy is involved, as in a multipolicy system, metapolicies must become explicit and flexible.

This paper illustrates metapolicies implicit in simple security policies, demonstrates how metapolicies can coordinate multiple security policies, and provides a foundation for future study of metapolicies.

EXECUTIVE SUMMARY

Definition

Metapolicies are policies about policies. They make the rules and assumptions about policies explicit rather than implicit and coordinate the interaction of multiple policies.

Functions

Metapolicies:

Describe policy structure and interrelationships;
Control policy additions or modifications.
Coordinate policies and subpolicies;
Establish policy precedence;
Resolve ambiguities;
Resolve policy conflicts;

This paper specifically identifies example metapolicies which accomplish each of the above functions. There are other functions as well. Dobson and McDermid, for example, identified a need for metapolicies which optimize, establish degrees of trust, and manage privileges and roles, as well as make policy assumptions explicit, and coordinate multiple policies.

Benefits

Metapolicies:

- Clarify security policies, including underlying assumptions, interactions and integration;
- Increase policy flexibility;
- Allow multiple policies in a system;
- Create a framework for complex security policies;
- Permit diverse and rich security policies;
- Permit tailored policy systems to match the legal and organizational policies of diverse clients.

Drawbacks

- The concept is unproven;
- They may add complexity to already complex systems;
- Secure systems may take even more time and money to design, develop and evaluate.

INTRODUCTION

OVERVIEW

Metapolicies, or "policies about policies", may become a powerful concept for developing the large, complex, and interrelated trusted systems that military, commercial and non-profit organizations need today. Metapolicies provide a framework for clarifying policies and for successfully coordinating security policies and subpolicies.

This paper develops the metapolicy concept first introduced in "Integrating Security Policies"[1] and expanded in "The Multipolicy Machine: A New Paradigm For Multilevel Secure Systems"[2]. As a prerequisite to tackling the problem of fitting multiple policies together with metapolicies, this paper explores the different kinds of metapolicies and defines their characteristics. It makes implicit metapolicies explicit, and demonstrates how metapolicies can be used to coordinate multiple security policies.

PRELIMINARY DEFINITIONS

Moffit and Sloman define **policies** as "the plans of an organization to meet its goals"[3]. **Subpolicies** are policies which contribute to a broader policy. **Security policies** are the plans of an organization to meet its security goals, often generalized as confidentiality, integrity, and availability. Those portions of the organization security policies which are implemented on the computer are called **automated security policies**[4]. Automated security policies traditionally comprise identification and authentication (I&A) policies, access control policies, audit policies, and backup and recovery policies, among others.

Although the U.S.A. Department of Defense Trusted Computer System Evaluation Criteria (TCSEC)[5] calls for a single

[1] Hosmer, Hilary H., "Integrating Security Policies", *Proceedings of the Third RADC Database Security Workshop, June 5-7, 1990, Castle, N.Y* ., MITRE MTP 385, May 1991.

[2] Hosmer, Hilary H., "The Multipolicy Machine: A New Paradigm for Multilevel Secure Systems", Standard Security Label for GOSIP: An Invitational Workshop, Gaithersburg, MD, NISTIR 4614, June 1991.

[3] Moffit, Jonathan D. and Morris S. Sloman, "The Representation of Policies as System Objects", Proceedings of the Conference on Organizational Computer Systems (COCS'91), Atlanta, Georgia, 5-8 November, 1991.

[4] Sterne, Daniel, "On the Buzzword Security Policy", Proceedings of the 1991 IEEE Computer Security Symposium on Research in Security and Privacy, Oakland, CA, May 20-22, 1991.

[5] Department of Defense Trusted Computer System Evaluation Criteria, DOD 5200.28-STD, December 1985.

system security policy, **multiple security policies** may be necessary if:

1) there is more than one security goal (for example, confidentiality and integrity);

2) the system serves diverse constituents each with its own goals and plans (for example, the multinational systems of the European Community will serve 12 or more sovereign nations);

3) the system is composed of separately evaluated pieces each with its own policy.

An information system which enforces multiple security policies is a **multipolicy system.**

THE NEED FOR MULTIPOLICY SYSTEMS

The world needs the ability to enforce multiple and diverse security policies in trusted computer systems. For example:

Hospitals must meet state, national and local integrity and privacy regulations as well as enforce professional rules and their own policies.

Multinational corporations must enforce the confidentiality rules and regulations of many nations and cultures as well as those of their own organization.

Multinational armed forces, such as those amassed in the Persian Gulf for Desert Storm, must work and communicate classified information across international and interservice security policy borders.

Software development environments must enforce three security policies, one for product development, another for product evaluation, and a third for product use.[6]

Banks must enforce multiple access control policies, including role-based policies[7] and Chinese Wall policies[8].

[6] Dobson, John, McDermid, John, "A Framework for Expressing Models of Security Policy", *IEEE Computer*, July 1989.

[7] Thomsen, D.J. "Role-based Application Design and Enforcement", Proceedings of the Fourth IFIP Workshop on Database Security, Halifax, England, September, 1990.

[8] Brewer, Dr. David F.C. and Dr. Michael J. Nash, "The Chinese Wall Security Policy", *Proceedings of the 1989 IEEE Computer Security Symposium on Security and Privacy*, Oakland, CA, 1989.

Current systems based on the TCSEC integrate multiple security policies poorly. The information security officer is expected to manually integrate multiple, possibly contradictory policies into a coherent system security policy. This is a difficult process, since each policy has its own source or owner, its own enforcement authorities, and its own evolutionary time frame.

A multipolicy machine which would enable a single computer to implement many security policies, even contradictory ones, was proposed in earlier work.[9] [10]. Metapolicies, a critical component of the multipolicy machine, are the focus of this paper.

METAPOLICIES

METAPOLICIES DEFINED

A metapolicy is a policy about policies. It may be either:

> 1) a set of rules about a single policy, specifying, for example, what kind of policy it is, the universe or domain to which the policy applies, who has the authority to change the policy, the procedure for changing policies, and the relationships to subpolicies;
>
> or
>
> 2) a set of rules for coordinating the enforcement of multiple policies, specifying, for example, the order in which multiple policies are enforced, and which results have precedence if a conflict in policies occurs.

Metapolicies make the rules about policy explicit rather than implicit.

Explicit metapolicies are not a new concept. For example, social clubs and other organizations often have a set of rules for the club and a separate set of by-laws which describes how the club rules are established and changed. The club rules are the club's policy, and the by-laws are the club's metapolicy.

[9] Hosmer, Hilary H. "The Multipolicy Machine: A New Paradigm for Multilevel Secure Systems", *Proceedings of the National Institute of Standards and Technology Workshop on Secure Labels*, Gaithersburg, Maryland, April 9-10, 1991.

[10] Hosmer, Hilary H.. "A Multipolicy Model: A Working Paper", *Proceedings of the Fourth RADC Workshop on Multilevel Secure Database Systems*, Little Compton, Rhode Island, June 1991.

However, security metapolicies are usually implicit and built into both hardware and software. For example, in the LOCK system, to get access to an object a user must meet the combined access control requirements of three separate policies: a standard MAC policy, a type enforcement policy, and an integrity policy[11]. A Boolean AND operation built into the hardware combines the results of the user's request to access the object under each of the three policies. This built-in metapolicy can't be changed to some other combination policy, such as OR or XOR. This immutability provides assurance that the metapolicy won't be changed, but at the cost of flexibility.

Most security metapolicies today are as invisible and immutable as the LOCK example. In the next section we explore how to make them explicit.

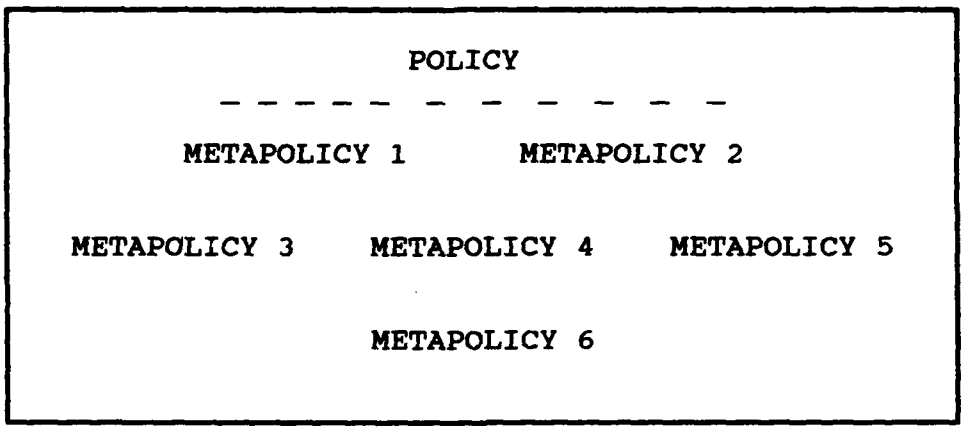


Figure 1

IMPLICIT METAPOLICIES

Metapolicies are inherent in existing security policies. To illustrate, Figure 2 on the next page is an adaptation of Bell and LaPadula's Simple Security Property. The author found six different metapolicies implicit in the Simple Security Property. These are illustrated on the following pages.

(Note that partial order and some other details have been deliberately omitted from this version of the Simple Security Property in order to focus on metapolicies.)

[11] Haigh, T. ACM SIGSAC presentation, NCSC Conference, Washington, D.C. October 3, 1991.

ADAPTATION OF SIMPLE SECURITY PROPERTY

```
POLICY NAME:      No Read Up
POLICY TYPE:      Access Control Subpolicy
AUTHORITY:        Secretary of Defense
CHANGE PROCESS:   Consultation with Armed Services

APPLICATION DOMAIN
     This policy applies to all computer systems
     containing USA military classified data.

INFORMAL STATEMENT OF POLICY
          No user or process representing a user may
          read data at a higher classification level
          than the user's login clearance level.

     EXCEPTIONS
          Users or processes with downgrade privilege
          are excepted.
     RELATED AUDIT POLICIES
          Security-relevant events must be auditable.
          Attempted violations must be auditable.
          Any violation must be audited and alarmed.
          Every use of downgrade privilege must be
               audited.
     OTHER RELATED POLICIES
          Users must identify themselves and be
          authenticated at login.
     PRECEDENCE RULES
          This policy has priority over any other
          access control policy.

FORMAL STATEMENT OF POLICY
          S   Subject:  User, process, active entity
          O   Object:  File, passive entity
          CR  CleaRance
          CL  CLassification

     May_Read (S, O)
       Begin
         If CR(S) >= CL(O)     \check simple security\
            then May_Read = YES
         Else
         If Downgrade(S)= YES     \downgrade privilege?\
            then May_Read = YES.
         If Audit (May_Read) = YES
            then write audit record
       End
```

Figure 2

In Figure 2, the implicit metapolicy components are:

1. A Policy Description Metapolicy

The names of the elements, the structure of the presentation, and the conventions of the policy description (such as both informal and formal policy statements) constitute a metapolicy framework that gives meaning to the elements of the policy much the way that data description metadata gives meaning to raw data elements in the database world.

2. A Policy Constraint Metapolicy

This metapolicy specifies the constraints put on the policy. These could include the application domain or time limitations on the policy due to expiration date, phased processing, or different day and nighttime policies. Other constraints might exempt certain users or roles from the policy, or require the policy to be executed in combination with another policy, for example.

3. A Subpolicy Interaction Metapolicy.

The policy portion in Figure 2 explicitly operates in concert with other policy portions, such as login, audit, downgrade, label interpretation, and application-specific access control policies. There is a hierarchy, implicit or explicit, implied with each of these, and many possible interactions which need to be defined.

SUBPOLICY STRUCTURE

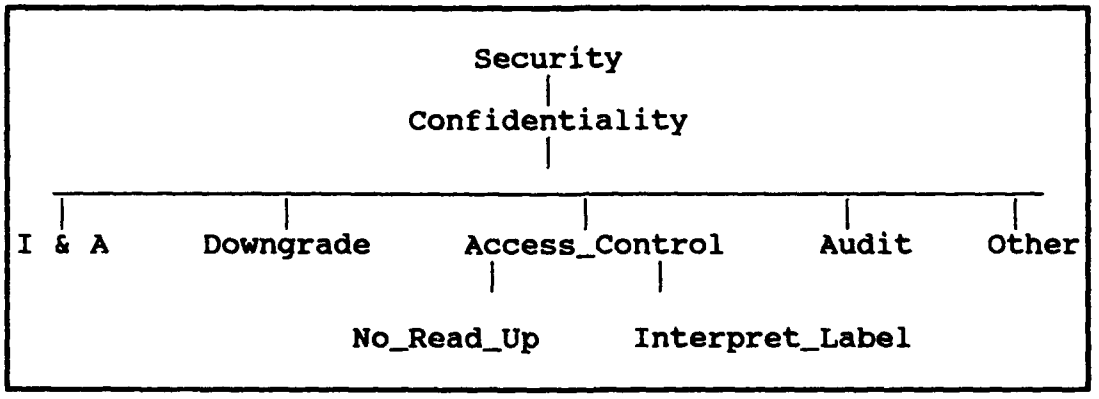


Figure 3

4. The Organization Control Metapolicy.

This metapolicy describes who owns the policy, who created the policy when, when the policy expires, whether the policy

can be renewed or modified, and what the processes are for distribution, renewal and modification. It may also include the legal status of the policy, the source of the policy (eg. Executive Order 12356) and in what documents the policy appears. The organization control metapolicy is critical in the real world, especially for policy conflict resolution, and an example policy is included in Figure 4.

5. Automated Information System Metapolicy.

This metapolicy is needed to control or describe the implementation of the policy in an automated information system. When general policy statements are turned into a site-specific policy instantiation, this metapolicy is the receiving area for items like constraints on implementation mechanisms (eg. MAC, DAC, encryption devices, or other) and requirements for configuration management and audit.

6. Multipolicy Coordination Metapolicy.

In a multipolicy machine, a security policy must interact with one or more security policies which may all claim precedence. This metapolicy coordinates all of the security policies as needed. This is a complex metapolicy, with many levels, domains, and implementation forms. An example is provided in Figure 6.

In summary, the Simple Security Property has several implicit metapolicies:

Policy Description Metapolicy
Policy Constraint Metapolicy
Subpolicy Interaction Metapolicy
Organization Control Metapolicy
Automated Information System Metapolicy
Multipolicy Coordination Metapolicy.

John Dobson and John McDermid, in their studies of security policies for software development environments[12], uncovered needs for several other metapolicies, although they didn't call them that. They identified needs for:

1) An Optimization metapolicy to resolve conflicting organization objectives, like producing high quality at low cost.

2) A Trust metapolicy to set up a framework permitting different levels and degrees of trust in different contexts.

[12] Dobson, John E., John A. McDermid, "A Framework for Expressing Models of Security Policy", *IEEE Computer*, July 1989.

ORGANIZATION CONTROL METAPOLICY EXAMPLE

UNDERLYING POLICY
Policy Name: No Read Up
Legal Status of policy: Mandated by federal law
Source of policy: Executive Order 123456

POLICY ORIGINS
Owners: DoD
Creator: Defense AIS Security Policy Center
Date Created: 1962
Expiration Date: 1992
Authors: John Smith, Sarah Jones
Reviewers: MITRE, Aerospace

APPROVAL PROCESS
Final Authority: President of the USA
Approving Organizations: US Department of Defense Army, Navy, Marines, Air Force, Coast Guard, Defense Intelligence Agency, DARPA, Joint Chiefs of Staff
Approval Sequence: Approving organizations give their approval in parallel, then it goes to the President

DISTRIBUTION: (Unlimited/Limited/Controlled)
Unlimited

POLICY USED IN:
Government Documents: DOD 654321, AF 802-456
Commercial Hardware: all MLS products
Commercial Software: all MLS products

RENEWAL
Renewal authorization: President of USA
Renewal terms: 3 to 10 years
Renewal Process: Service and JCS approval

MODIFICATION
Authorization for modification: President of USA
Policy modifier: US AIS Security Policy Center
Process for modification: Review by all services
Last Date Modified: 1985

PUBLICATION DATA:
Publisher: DoD Publications Center
Document: Military AIS Security Policy

Figure 4

3) A Privilege metapolicy for tracking each software development tool's security properties.

4) A Role metapolicy to define the rules of role-based policies.

5) An Assumptions metapolicy to clarify the assumptions underlying the policies.

Dobson and McDermid also encountered the need for a framework which would coordinate multiple policies encountered in a software development tool system.

COORDINATING MULTIPLE POLICIES

INTRODUCTION

"Multipolicies are the norm, not the exception" says Eric Leighninger, a Senior Computer Scientist at Dynamics Resource Corporation, speaking of his years of experience with military systems. Many computer security scientists have wrestled with the problems of how to coordinate multuple security policies. This paper does not review all the approaches taken, but focuses on a few archtypes.

SEPARATION

One way to coordinate multiple policies is to keep them physically separate, operating in completely distinct domains without communication. John Rushby's Separation Machine, implemented by Amdahl[13], provides a way for one machine to run seven separate security policies in parallel.

The National Computer Security Center's *Trusted Database Interpretation of the Trusted Computer System Evaluation Criteria* (TDI) describes the principles of TCB subsets, in which multiple policies such as mandatory access control (MAC) and disretionary access control (DAC), operate independently by operating on different logical objects.[14] The TDI proposes logical rather than physical separation.

If policies come into conflict with one another, such as confidentiality and integrity or MAC and DAC, there must be strategies for resolving the conflicts. The most popular are precedence (MAC has priority over DAC), parallel resolution with precedence, and setting defaults.

[13] Amdahl Corporation, Multiple Domain Feature: General Information Manual, CA, 1989.

[14] National Computer Security Center, Trusted Database Management System Interpretation of the Trusted Computer System Evaluation Criteria, April 1991.

PRECEDENCE

Several different metapolicy frameworks are possible for coordinating multiple policies. Moffet and Sloman[15] favor precedence ordering, appropriate when:

a) policies conflict
   eg. One policy requires something, the other forbids it.
   eg. One policy invalidates actions done under another policy

b) two actions are incompatible

c) there is competition for scarce resources.

Traditionally, precedence is hard-coded or hard-wired. It is built into most systems, for example, that MAC has priority over DAC. This has the advantage of security at the price of flexibility.

Precedence can be implemented in other ways. It can be computed. For example, policies can assigned priority levels which indicate which policy prevails when any two or more conflict. Or precedence can be looked up. For example, tables can be provided listing precedence for every possible pair of policies. LaPadula uses tables to establish precedence in his rule-based multiple policy model[16]. (To improve performance, these tables should be cached in high-speed memory when implemented.)

PARALLEL WITH PRECEDENCE

To resolve multiple conflicts quickly and systematically, parallel processing of separate policies is promising. Figure 5 adapts Marshall Abrams and Michael Joyce's unpublished proposal to extend the International Standards Organization (ISO) Access Control Framework to include parallel policy processing and metapolicies.

The conflict resolution process in Figure 5 works as follows:

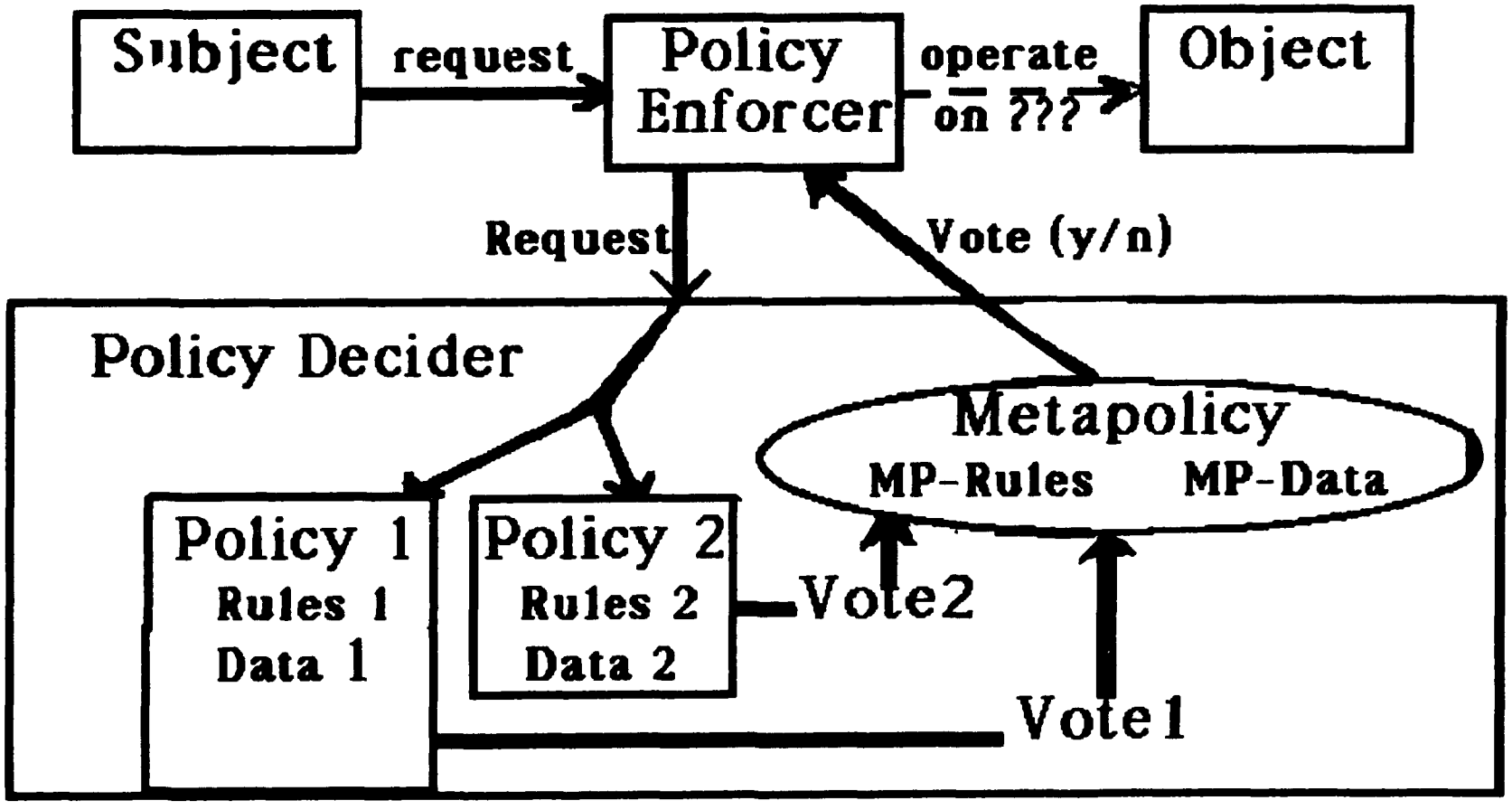
1) The 'Subject' wants to operate on the 'Object', but the request must be mediated by the 'Policy Enforcer'.

[15] Moffet, Jonathan D. and Morris S. Sloman, "The Representation of Policies as System Objects", SIGOIS Bulletin vol 12, nos 2&3 pp 171-184.

[16] LaPadula, L.J. "A Rule-Base Approach to Formal Modeling of a Trusted Computer System", MITRE M91-021, Aug. 1991.

MULTIPLE POLICY FRAMEWORK

Figure 5. Adapted from a diagram by M. Abrams and M. Joyce.



2) The Policy Enforcer turns the request over to the 'Policy Decider', which consists of multiple Policy Decision-Makers, one for each policy implemented by the system. Each Policy Decision-Maker consists of rules and decision data, and they operate in parallel.

3) Using rules and decision data to determine whether the request is acceptable under its policy, each policy decision-maker sends a 'Yes' or 'No' or 'Undecided' vote to the metapolicy.

4) The votes of all the individual policies (Vote 1 and Vote 2 in this example) are then combined by the Metapolicy facility according to its rules and data.

5) The resulting 'Yes' or No 'vote' is sent back to the Policy Enforcer which permits or denies the requested operation.

The conflict resolution process is simple and elegant, no matter how many different policies are included.

SELECTING DEFAULTS

In other situations of conflict, ambiguity makes a simple precedence order impossible. Holden, in his work on management policy[17], describes multiple causes for policy ambiguity. These include:

a) multiple sources of policy;

b) under-specification of policy;

c) inadequate information to resolve policy references.

Metapolicies become critical in these ambiguous cases. In particular, metapolicies permit user specification of a default option.

For example, if an access request policy conflict is undecidable, there are three choices:

(1) Reject the request (the conservative approach traditionally preferred by the military and the intelligence community);

(2) Permit access (the liberal approach traditionally preferred by the academic); or

[17] Holden, D.B., An Exploration of the Nature of Management Policy, BSPRJT/5165/harw/T2.1/1_0, AIA Industrial Technology, Harwell Laboratory, Oxfordshire, UK. 5 Feb 1991.

(3) Defer the decision until it is decidable (the safe approach preferred by the bureaucrat).

Fortunately, the process of defining the policy interactions and the desired metapolicy will reduce the amount of ambiguity.
If the system vendor chooses the default for the remaining ambiguous cases, the conservative approach will most likely be taken.

Since even a conscientious vendor can't always anticipate which default will be preferable to every customer, it is critical that the end-user have the flexibility to choose what to do in ambiguous situations. The users' rules could be specified at system generation time, at start-up, or even during operations if changing circumstances require it. Providing end-user flexibility is critical to the metapolicy concept and a major motivator for using it.

CHARACTERISTICS OF METAPOLICIES

Generalizations

From the single policy and multiple policy examples given above, it is possible to generalize about the characteristics of metapolicies.

1. Their primary objective seems to be providing control, for the organization, for the AIS, and for the security subsystem. The importance of this control function is confirmed by the work of McDermid and Hocking[18] who identify three pages of control objectives for security policies.

2. Metapolicies may contain data or rules and may vary in complexity from a single value to an elaborate modular and layered data structure.

3. There appear to be multiple metapolicies for every security policy.

4. Implicit metapolicies aren't obvious, and there seems to be an art to making the implicit explicit.

5. User flexibility is needed. The privilege of establishing some metapolicy rules and data must belong to the user security officer as well as the trusted system vendor.

[18] McDermid, John, Ernest Hocking, "Security Policies for Integrated Project Support Environments", *Database Security III: Status and Prospects*, North Holland, 1989.

More Generalizations

In addition to the examples above, the author studied five applications, including a multinational bank (commercial), a hospital (non-profit), a DoD network (military), a multinational alliance (military), the Chinese Wall Policy (research)[19] and LaPadula's formal GFAC model[20] (research) which incorporates five security policies.

Several additional metapolicy characteristics were uncovered in the applications.

6. Metapolicies are layered, corresponding to the layers of the organization, the layers of the computer system and the layers of security policies. Network metapolicies, node metapolicies, and application-specific metapolicies are a few examples. In a federated system, metapolicies would be required at federal, local, and intermediate levels.

7. There are general rules which hold for all situations, and there are sets of rules which apply to certain situations, but not others. There are bilateral agreements which apply only to two parties. Metapolicies must be able to handle all these possibilities: general rules, group or subset rules, and individual rules.

8. Like any security policy, all metapolicies must be protected from tampering or interference. Changes or additions must be audited. If stored on hardware or firmware, validation of the correct operation of the hardware or firmware must be provided. In short, all the requirements that apply to any Trusted Computing Base (TCB), apply to metapolicies, since those portions which are implemented in a computer system become a component of the TCB.

9. Metapolicies are most critical at security policy domain interfaces. For example, if data labelled for one policy domain must be transferred to another policy domain, there will be a policy about policies, or metapolicy, describing the rules for this transfer.

10. Security policies can change over time. Metapolicies, by providing control data like that in Figure 3, provide a foundation for conscious and careful evolution. Metapolicies provide support for the variety of security policy forms and control for their evolution.

[19] Brewer, Dr. David F.C. and Dr. Michael J. Nash, "The Chinese Wall Security Policy", *Proceedings of the 1989 IEEE Computer Security Symposium on Security and Privacy*, Oakland, CA, 1989.

[20] LaPadula, Leonard, " A Rule-Base Approach to Formal Modeling of a Trusted Computer System", MITRE, M91-021, August 1991.

A list of metapolicy functions was extracted from the applications studied and the characteristics uncovered. This list of metapolicy functions is explained below.

METAPOLICY FUNCTIONS

Sequence

One of the key metapolicy functions is to specify the sequence in which security policies will be enforced. In a hospital serving patients from multiple states, the system administrator must decide which state's laws come first when more than one state is involved. Some strategies might be: "Most restrictive states first", or "Hospital's state, then patient's state, then other involved states in alphabetical order". In a hierarchical structure, the policy might be "top-level first, then succeeding levels as long as each meets the criteria".

Managing Contradictions

When there is more than one policy, contradictions among policies are inevitable. The metapolicy must describe how to handle these contradictions. Sometimes the metapolicy will simply specify which policy predominates, like the words on government papers which read, "In case of contradiction, this document prevails." In a hierarchical system, the dominant policy would prevail unless exceptions were specified. The ability to specify predominance may handle Leslie Chalmers' concern that government systems emphasize confidentiality while commercial systems emphasize integrity[21]. Other times, there may be special conditions, such as a threat alert, which determine which policy predominates.

Multiple Policy Organizations

The metapolicy must be able to handle hierarchy, network, parallel and other organizations of policies. In the DoD, for example, the metapolicy will specify that the DoD policy is enforced first, then the Air Force (AF) policy, then the Air Force Base (AFB) policy. If any higher-level policy is not satisfied, the policy-check fails and there is no need to check the subordinate policies. However, the rules are different for the Defense Intelligence Agency (DIA) which is not subordinate to the DoD policy.

[21] Chalmers, Leslie, "An Analysis of the Differences Between the Computer Security Practices in the Military and Private Sectors", *Proceedings of the 1986 Symposium on Security and Privacy*, 1986, pages 71-74.

Flexibility

It is clear that there are many different administratively-imposed policies. Some are legally-mandated, others organizationally-required, some derived from standards, some driven by computer norms. It is very important that there be flexibility to incorporate all these different policies. Currently, one only gets the policy which comes with the system and the ability to modify a policy lattice defining levels and categories. It is very difficult to include the specific needs of an organization in such an inflexible structure.

In addition to including a variety of administrative policies, it is important that these policies be modifiable by the appropriate authorities or their representatives, and only by them. The system security officer may implement the policy changes on the system, but the policies themselves must be modified by the responsible authorities.

Enforcement

Like access control policies, metapolicies can be specified and enforced globally and locally. A metapolicy indicator could be attached to each object along with the other security policy indicators to be referenced whenever the object is referenced. A metapolicy might be employed at the node level, so that objects entering the node are required to take on a particular node security policy as well as their own security policies. Finally, many metapolicy rules should be applied at the global level, such as when security policies may be updated and by whom.

Adding and changing policies

Metapolicies would specify whether policies may be added, deleted, or modified and by whom. Policies may accumulate as objects enter multiple policy domains, and the problems of sequence, dominance, etc. must be resolved. One metapolicy rule might be that security policies can be added to objects, but not removed from objects except by the security policy originator. Site security administrators would normally be responsible for making modifications to security policies, but changing security policy is so critical, there may be a requirement that two security officers make any changes jointly.

Layers of Metapolicies

There needs to be some differentiation between metapolicies of vastly different scope and significance. There will be metapolicies about metapolicies, for example, as well as metepolicies about policies. Metapolicies must be well-

structured so that it is easy to see what prevails over what.

Protection from Tampering

Like the rest of the TCB, metapolicies are critical to the functioning of the security system and must be protected from tampering.

Operate at Interface Boundaries

Metapolicies are critical at the boundaries between systems. These interface metapolicies will determine if data can cross the boundary between systems based upon a complete set of rules for determining if this is possible. If necessary, the metapolicy will map from one security policy to another. This is a major topic and is left to future work.

Risks When Sharing

It is clearly risky to share policies and metapolicies with other systems. An enemy might find flaws which could be exploited or make malicious modifications. Yet, sharing with other systems is what makes it possible for each system to enforce multiple policies.

Audit and Inspections

Changes to metapolicy are security-relevant events, just like changes to policy. They should be implemented only by the system security officer or a representative. Major changes might require the two-man rule. All changes to metapolicy should be audited. In systems that share policies, there should be a periodic (but surprise) configuration audits to verify that there have been no unauthorized changes to the shared policy.

Knowing what the functions of metapolicies are, how does one go about developing metapolicies and implementing them?

FUTURE WORK

Policies are a major topic of study, as the substantial work devoted to this subject at management and organizational conferences shows. However, work on multiple policies and metapolicies is only just beginning. Metapolicy is a new concept to computer security, and this paper, the first one devoted entirely to the topic, only scratches the surface.

Both multiple policies and metapolicies need to be studied in more depth. Currently, most security researchers mean MAC/DAC or confidentiality/integrity when they think of multiple policies. Their horizons need to be expanded to incorporate many other kinds of policies, including application policies.

More work needs to be done on integrating security policies together. The author outlined some of the problems in "Integrating Security Policies" and plans to devote a future paper to fitting multiple policies together with metapolicies.

Researchers must find ways to permit the end-user security officer to shape system policy to meet his/her needs. This paper mentioned installing policy at sysgen or start-up time. Currently, DAC is the vehicle for enforcing end-user's policies, but the individual discretion inherent in DAC mechanisms makes them inappropriate for site administratively-controlled policies. A previous paper, "The Multipolicy Machine: A New Paradigm For Multilevel Secure Systems", suggested a model of multiple interacting policies where the user site policies would complement but not interfere with the sytem policies. More work is needed here.

Theory development

Metapolicy theory needs expansion and development. Metapolicies need to be explored in more detail and specified in a formal specification language like BNF or ASLAN. Metapolicies need to be modelled and prototyped, and a taxonomy of metapolicy types needs to be developed. Finally, the term metapolicy needs a precise and formal definition.

There are so many possible combinations of policy rules, strategies must be developed to group policies and metapolicies. To determine what should be in the metapolicies, several techniques can be used:

> *Policy inspection* is a useful first step because it provides intuitive ideas about what is needed.
>
> *Abstraction and generalization* make it possible to solve a problem once rather than many times. For example, it may be possible to generalize a limited set of formats for descriptive, control, and other types of metapolicies.

Graphic techniques, such as entity-relationship diagrams[22] and trees, can help visualize the metapolicy relationships, groupings and interactions. For example, in their paper on formalizing policy managment, Sibley, Michael and Wexelblat use data flow diagrams, structural diagrams, activity-role charts[23] and Petri nets[24] to visualize policy interactions[25]. Backus Naur Forms (BNF structures)[26], a higher-order description method used to specify Algol and other programming languages, could be useful in specifying policies and metapolicies as well.

Knowledge-engineering strategies from expert systems will be useful in making the implicit explicit and in defining policy and metapolicy rules.

Formal modeling provides a rigorous analysis and clarification of implicit policies and interactions. LaPadula's GFAC model is an example of this[27].

Implementation

Metapolicies must be implemented at multiple levels. There are global rules, domain rules, and rules associated only with one or a few policies.

Metapolicies can be implemented in several ways:

Databases are excellent at preserving data and relationships. An active policy/metapolicy database, such as the *active data dictionaries*, would enforce as well as store policy. Sibley, Michael, and Wexelblat analyze the feasibility of three variations of an active data dictionary for policy: Procedure Enforced Policy, State Condition Enforced Policy, and Procedure-

[22] Chen, Peter. The Entity-Relationship Approach to Logical Data Base Design, Q.E.D. Information Systems Inc. Wellesley, MA 1977.

[23] Holt, A.W. Ramsey H.R. and Grimes, J.D. "Coordination System Technology as the Basis for A Programming Effort", Electrical Communication, Vol. 57, No. 4, 1983.

[24] Murata, T. ,"Petri Nets: Properties, Analysis, and Applications," Proceedings of the IEEE 77, 541-580.

[25] Sibley, Edgar, James B. Michael, and Richard Wexelblat, "An Approach to Formalizing Policy Management", CECOIA2-Proceedings of the 2nd International Conference on Economics and Artificial Intelligence, Pergamon Press, Oxford, England, 1991.

[26] Jay, Chris, "A BNF Compiler For Prolog (Backus Naur Form description of programming languages), AI Expert Vol V6, Jan 1991

[27] Lapadula, L.J., "Formal Modeling in a Generalized Framework for Access Control", Proceedings of the Computer Security Foundation Workshop III, June 1990.

Restriction-and-Checking Enforced Policy[28]. Their strategies could apply to metapolicy as well.

In a different approach to using databases, Moffet and Sloman's work on the "Representation of Policies as Systems Objects"[29] is a promising technique which can be applied to metapolicies as well as policies. It makes inheritance, the ability to query, and other features of *the object-oriented database model* available to the policy/metapolicy model.

Rule-based systems like those advocated by John Page[30], Jody Heaney, Marshall Abrams[31], Leonard LaPadula and others, are an excellent choice. Figure 5 shows how metapolicies fit into a rule-based system, and that the metapolicies can be stored in a rule-base as well.

Finally, metapolicies can be physically implemented in all the ways that other policies are implemented. If there is sufficient standardization, metapolicies can be coded on ROM chips or installed on co-processors. It makes sense to employ parallel processors so that the metapolicies can be processed along with the policies.

CONCLUSION

Security policies are much more sophisticated than originally thought when the DOD security policy model dominated the field. Many researchers have looked for a framework which would help develop and manage these sophisticated policies. Multipolicy machines with metapolicies promise a conceptually elegant framework for managing sophisticated security policies. However, there is much research and development to be done before the promise becomes a reality.

This paper provided an introduction to and an overview of the Metapolicy concept. It illustrated that metapolicies can clarify underlying policy assumptions and relationships and facilitate expression of the variety, richness, and

[28] Sibley, Edgar, James B. Michael, and Richard Wexelblat, "An Approach to Formalizing Policy Management", CECOIA2-Proceedings of the 2nd International Conference on Economics and Artificial Intelligence, Pergamon Press, Oxford, England, 1991.

[29] Moffett, Jonathan D. and Morris S. Sloman, "The Representation of Policies as System Objects", *Proceedings of the Conference on Organizational Computer Systems (COCS'91)* Atlanta, Georgia 5-8 November, 1991.

[30] Page, John, Jody Heaney, Marc Adkins, and Gary Dolsen, "Evaluation of Security Model Rule Bases", Proceedings of the 12th National Computer Security Conference, 10-13 October 1989.

[31] M.D. Abrams, K.W. Eggen, L.J. La Padula, and I.M. Olson, "A Generalized Framework for Access Control: An Informal Description," *Proceedings 13th National Computer Security Conference*, Washington, D.C. October 1990.

multiplicity of security policies. It illustrated how metapolicies might permit the controlled interaction of policies and subpolicies, making complex policy systems possible.

Metapolicies, or 'policies about policies', may become a powerful concept for coordinating the multiple, complex, and interrelated security policies that military, commercial, and non-profit organizations need today.

***

ACKNOWLEDGEMENTS

Grace Hammonds of AGCS reviewed the document and made many constructive suggestions. J. Bret Michael of IDA, who is earning a PhD studying security policy, critiqued the paper and contributed an invaluable set of recent technical papers. Marshall Abrams of MITRE provided his yet unpublished work in the multipolicy area. Eric Leighninger of Dynamics Research Corporation contributed the suggestion that policies and metapolicies be represented with the Backus Naur Form (BNF) and developed the slogan "Multipolicy Systems Are the Norm". Victoria Ashby of MITRE created the opportunity to present the multipolicy concept in two ACM SIGSAC workshops at two major conferences. Rowena Chester of Martin Marietta Energy Systems coordinated the ACM SIGSAC workshop on Dec. 3, 1991 in San Antonio, Texas, where this paper was first presented. Thank you all.

REFERENCES

Abrams, M.D., K.W. Eggers, L.J. La Padula, and I.M. Olson, "A Generalized Framework for Access Control: An Informal Description," *Proceedings 13th National Computer Security Conference*, Washington, D.C. October 1990.

Amdahl Corporation, *Multiple Domain Feature: General Information Manual*, CA, 1989.

Bell, D.E. and L.J. LaPadula, "Secure Computer Systems, Unified Exposition of Multics Interpretation", MTR-2997, Rev. 1, The MITRE Corporation, Bedford, MA 1976.

Benzel, Terry C. Vickers, "Formal Policies for Trusted Processes", position paper, Seventh Annual Computer Security Applications Conference, December, 1991.

Brewer, Dr. David F.C. and Dr. Michael J. Nash, "The Chinese Wall Security Policy", *Proceedings of the 1989 IEEE Computer Security Symposium on Security and Privacy*, Oakland, CA, 1989.

Chalmers, Leslie, "An Analysis of the Differences Between the Computer Security Practices in the Military and Private Sectors", *Proceedings of the 1986 Symposium on Security and Privacy*, 1986, pages 71-74.

Chen, Peter, The Entity-Relationship Approach to Logical Data Base Design, Q.E.D. Information Systems Inc. Wellesley, MA 1977.

Department of Defense Trusted Computer System Evaluation Criteria, DOD 5200.28-STD, December 1985.

Dobson, J.E. and J.A. McDermid, "Security Models and Enterprise Models", *Database Security, II Status and Prospect*, North Holland, 1989.

Dobson, John, McDermid, John, "A Framework for Expressing Models of Security Policy", *IEEE Computer*, July 1989.

Executive Order 12356, *National Security Information*, 6 April 1982.

Haigh, T. ACM SIGSAC presentation, NCSC Conference, Washington, D.C. October 3, 1991.

Holden, D.B. "An Exploration of the Nature of Management Policy", ESPRIT,/5165/harw/T2.1/1_0, AEA Industrial Technology, Harwell Laboratory, Oxfordshire, UK 5 February 1991.

Holt, A.W. Ramsey H.R. and Grimes, J.D. "Coordination System Technology as the Basis for A Programming Effort", *Electrical Communication*, Vol. 57, No. 4, 1983.

Hosmer, Hilary H. "Integrating Security Policies", *Proceedings of the Third RADC Database Security Workshop June 5 - June 7, 1990, Castile New York*, MITRE MTP 385, May 1991.

Hosmer, Hilary H. "The Multipolicy Machine: A New Paradigm for Multilevel Secure Systems", *Proceedings of the National Institute of Standards and Technology Workshop on Secure Labels*, Gaithersburg, Maryland, April 9-10, 1991.

Hosmer, Hilary H.. "A Multipolicy Model: A Working Paper", *Proceedings of the Fourth RADC Workshop on Multilevel Secure Database Systems,* Little Compton, Rhode Island, June 1991.

Jay, Chris, "A BNF Compiler For Prolog (Backus Naur Form description of programming languages), *AI Expert,* Vol 6, Jan 1991

LaPadula, L.J. "A Rule-Base Approach to Formal Modeling of a Trusted Computer System", MITRE M91-021, Aug. 1991.

LaPadula, L.J., "Formal Modeling in a Generalized Framework for Access Control", Proceedings of the Computer Security Foundation Workshop III, June 1990.

Matley, Ben G. and Thomas A. McDannold, "National Computer Policies", IEEE, Washington D.C., 1987.

McDermid, John, Ernest Hocking, "Security Policies for Integrated Project Support Environments", *Database Security III: Status and Prospects*, North Holland, 1989.

Moffett, Jonathan D. and Morris S. Sloman, "The Source of Authority For Commercial Access Control", *IEEE Computer,* February 1988.

Moffett, Jonathan D. and Morris S. Sloman, "The Representation of Policies as System Objects", *Proceedings of the Conference on Organizational Computer Systems (COCS'91)* Atlanta, Georgia 5-8 November, 1991.

National Computer Security Center, *Trusted Database Management System Interpretation of the Trusted Computer System Evaluation Criteria*, April 1991,

Obal, Supreme Allied Commander Atlantic James, William Grogan, "A Case Study for the Approach To Developing a Multilevel Secure Command and Control System", *Proceedings of the 14th National Computer Security Conference*, Washington, D.C.,October 1991.

Price, Lt. Col William, Michael E. O'Neill, Frank B. White, "Accreditation Strategy for the Air Force Satellite Control Network (AFSCN) *Proceedings of the 14th National Computer Security Conference*, Washington, D.C.,October 1991.

Sibley, Edgar, James B. Michael, and Richard Wexelblat, "An Approach to Formalizing Policy Management", *CECOIA2-Proceedings of the 2nd International Conference on Economics and Artificial Intelligence*, Pergamon Press, Oxford, England, 1991.

Sibley, Edgar H., James Bret Michael, and Richard Wexelblat, "Use of an Experimental Policy Workbench: Description and Preliminary Results", *Proceedings of the IFIP TC11.3 5th Working Conference on Database Security*, North-Holland, Amsderdam, 1991.

Sterne, Daniel, Martha Branstad, Brian Hubbard, Barbara Mayer, Dawn Wolcott, "An Analysis of Application Specific Security Policies", *Proceedings of the 14th National Computer Security Conference*, Washington, D.C.,October 1991.

Thomsen, D.J. "Role-based Application Design and Enforcement", *Proceedings of the Fourth IFIP Workshop on Database Security*, Halifax, England, September, 1990.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	Dec. 3, 1991	Interim

4. TITLE AND SUBTITLE
METAPOLICIES I

5. FUNDING NUMBERS
C# F19628-91-C-0157

6. AUTHOR(S)
Hiliary H. Hosmer

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Data Security, Inc.
58 Wilson Rd.
Bedford, MA 01730

8. PERFORMING ORGANIZATION REPORT NUMBER
DSI-010-91

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Electronic Systems Division/AVS
Air Force Systems Command, USAF
Hanscom AFB, MA 01731-5320

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
To be published in ACM SIGSAC Review 1992

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unlimited availability

This document has been approved for public release and sale; its distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

Metapolicies, or "policies about policies", may become a powerful concept for developing the large, complex, and interrelated trusted systems that military, commercial and non-profit organizations need today. Metapolicies provide a framework for clarifying policies and for successfully coordinating security policies and subpolicies.

When there is only one security policy, metapolicies tend to be implicit, embedded, and fixed. When more than one security policy is involved, as in a multipolicy system, metapolicies must become explicit and flexible.

This paper illustrates metapolicies implicit in simple security policies, demonstrates how metapolicies can coordinate multiple security policies, and provides a foundation for future study of metapolicies.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Multipolicy, Security Policy, Metapolicy	26
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL