Internet Draft
draft-ietf-ipsp-msme-00.txt
Expires May, 2002

M. Condell, BBN

November 14, 2001

Multidimensional Security Policy Management and Enhancements for IP Security Policy

Status of this Memo

Copyright Notice

Abstract

This document describes the requirements and architecture for supporting security policy resolution among a coalition of partners. It then discusses how solutions necessary for the coalition resolution problem may be utilized to improve the on-going development of an IPSP solution.

Table of Contents

1. Introduction

This document describes the requirements and architecture for the Multidimensional Security Management and Enforcement (MSME) system. MSME provides a means for members of a dynamic coalition with limited trust between them to negotiate mutually agreeable policies that enable mission relevant communications. MSME's solution allows each member to maintain its own internal policy requirements, policy languages, and policy management systems while enabling them to exchange and resolve policies with other members of the coalition.

While it may be early for the IPSP working group to design solutions beyond IPsec policy resolution between two hosts and any intervening gateways, there are several benefits to looking at the requirements and potential solutions for more complex security policy management environments. It exposes issues with, and presents possible solutions to, generalizing the IPSP work to security protocols other than IPsec.

It may also help the working group to design a solution that is more extensible to other security policy management needs.

The next section describes the MSME system motivations, requirements, and architecture. It is followed by a discussion of some of the ideas that may be incorporated into the IPSP work.

1.1 Definitions

The following terms are used throughout this document, in addition to the terms defined for general policy terminology [TERM].

Coalition

> A coalition is a group of administrative entities (e.g. companies, countries) that work together to achieve a defined objective (mission). The coalition will have specific communications requirements necessary to accomplish the objective.

Partner

> A partner is an administrative entity that participates in a coalition.

Keywords "MUST", "MUST NOT", "REQUIRED", "SHOULD", "SHOULD NOT" and "MAY" that appear in this document are to be interpreted as described in [Bra97].

2 MSME

The MSME system is being developed under DARPA contract [Contract Number F30602-00-C-0062].

2.1 Motivation

Policy management, while fundamental for good network security, is a complex task. It is difficult in the IPSP world where two end-hosts are trying to communicate, but is even more complex when groups are trying to communicate. Coalitions are a particularly difficult environment for policy management. There may be a large number of partners that need to share policy information. Partners may join or leave the coalition, each change possibly affecting the policy that must be enforced. Each partner may have different management elements, including different policy languages, policy distribution protocols, and policy authoring tools. Each partner may even implement a different set of security protocols.

Despite the heterogeneity of the coalition environment, it is necessary to be able to determine if the coalition can implement the security policies necessary to carry out its mission. This requires that the coalition partners be able to exchange their policies, determine a common set of policies among the partners, and insure that all the communication requirements for the coalition can be met by all

the partners. If the requirements cannot be met, they must be flagged so an administrator can fix them before they interfere with the mission.

Additionally, once the policies are in place, partners will want to ensure that the policies are being used and enforced correctly. Each partner requires some level of monitoring to verify that it and other partners are correctly enforcing the agreed upon policies. How much information may be obtained from monitoring will depend upon where monitoring probes are allowed to be placed, which, in turn, affects what information is available to monitor.

2.2 Overview

The MSME architecture consists of a high-level policy description language; protocols for transferring the policy between partners; and algorithms for ensuring the internal consistency of the high-level policy description, resolving the policies between partners, and verifying that the resolved policy does not violate the policies of a particular partner.

When a coalition forms, membership changes, or one or more partners' policies change, each partner creates a policy level agreement (PLA). The PLA contains high-level policy requirements and mappings from the abstract requirements to concrete implementations that the partner supports. Each partner checks that their PLA is self consistent and that their local policies do not conflict with the high-level requirements. If there is a conflict or inconsistency, human intervention is required to correct the problem.

The partners then exchange their PLAs. Depending on the coalition, partners may either exchange their PLAs with each other, or they may hand them to a central server. The exchange must be authenticated and may require confidentiality. Each partner, or a central server, as agreed to by the coalition, must then resolve the PLAs. The resolution process determines which concrete policies the coalition members agree to use to implement the policies specified in their PLAs. PLAs may be resolved by subsets of the coalition partners to allow services among the subsets that do not involve the rest of the coalition. The resolution process produces a resolved PLA (RPLA). The RPLA will have to be distributed to each partner if the resolution process was centralized.

When each partner receives or computes the RPLA, it must reconcile it with its local policies to verify that it does not violate its local policy. Additionally, the reconcilliation process can detect policy rules that were not included in the RPLA because other coalition members could not support them and may require manual intervention if those communications must be supported. If the resolution process is executed properly and no partner misbehaves, the reconciliation process should not detect any rule violations, however, it is a necessary step to ensure there were no problems.

The dynamic coalition environment will produce a very complex set of policies, so it will be difficult to determine a priori if the correct policies were negotiated. Monitoring can help determine if the correct policies are being used and that policy enforcement points are

using them correctly. The power of monitoring will be limited by where partners will allow monitoring probes to be placed, both physically and because most communications will be encrypted.

Partners may have to take into account that policy may be defined by different people at different levels. The high-level policy may be defined by someone who established the coalition and its communication requirements, but does not know how the requirements get implemented. Similarly, bindings may be created by someone who understands the low-level policies, but did not determine the high-level policy.

2.3 Requirements

This section will discuss some of the main requirements for the MSME system.

* MSME MUST allow each partner to internally implement its own policy languages, policy storage and policy distribution mechanisms. However, MSME MAY impose requirements on them.

* MSME MUST provide a mechanism for partners to exchange their policies and to resolve policy information using a common protocol. The policies must be securely communicated, including authentication and integrity checks. MSME SHOULD have a mechanism by which coalition partners can associate security policy rules with specific coalition partners.

* MSME SHOULD support both private peering (sub-coalitions) and partial (or abstracted) sharing of internal policy information. Some partners may want to keep portions of their policy private from other partners. Obviously, those rules of their policy relevant to the policy resolution need to be released to other partners.

* MSME MUST provide a way to determine whether a specific communication is permitted by the current policy agreement between coalition partners before the communication is attempted.

* MSME MUST provide a way to establish and identify, at any time, the set of authorized coalition partners and the entities that they have authorized to engage in coalition activities.

* MSME MUST support security services implemented by multiple security protocols (e.g., IPsec, TLS), and compositions thereof. Therefore, MSME SHOULD support a security abstraction layer that can map (high-level) policy intent to different (low-level) security policy data models. Both the representation (language) and exchange (protocol) of this abstraction must be supported.

* MSME MUST provide a way to monitor policies while they are in use

to confirm that the correct policies are installed in the
enforcement points, that they are being applied correctly, and that

the policy decision points (PDPs) and policy enforcement points (PEAs) are behaving correctly. There must be a way to monitor both the coalition policies and the local policies of the PDPs and PEAs.

2.4 Architecture

Figure 1 illustrates the components of the MSME system. The partner-dependent components are not defined as part of the MSME system, however, the system interacts with those components and may impose requirements on them. Intra-partner components represent components fully contained within a partner. Inter-partner components which are not partner-dependent must exist as part of the MSME system, but don't have to directly interoperate with other partners. Other components are part of the MSME system and must be interoperable between partners.

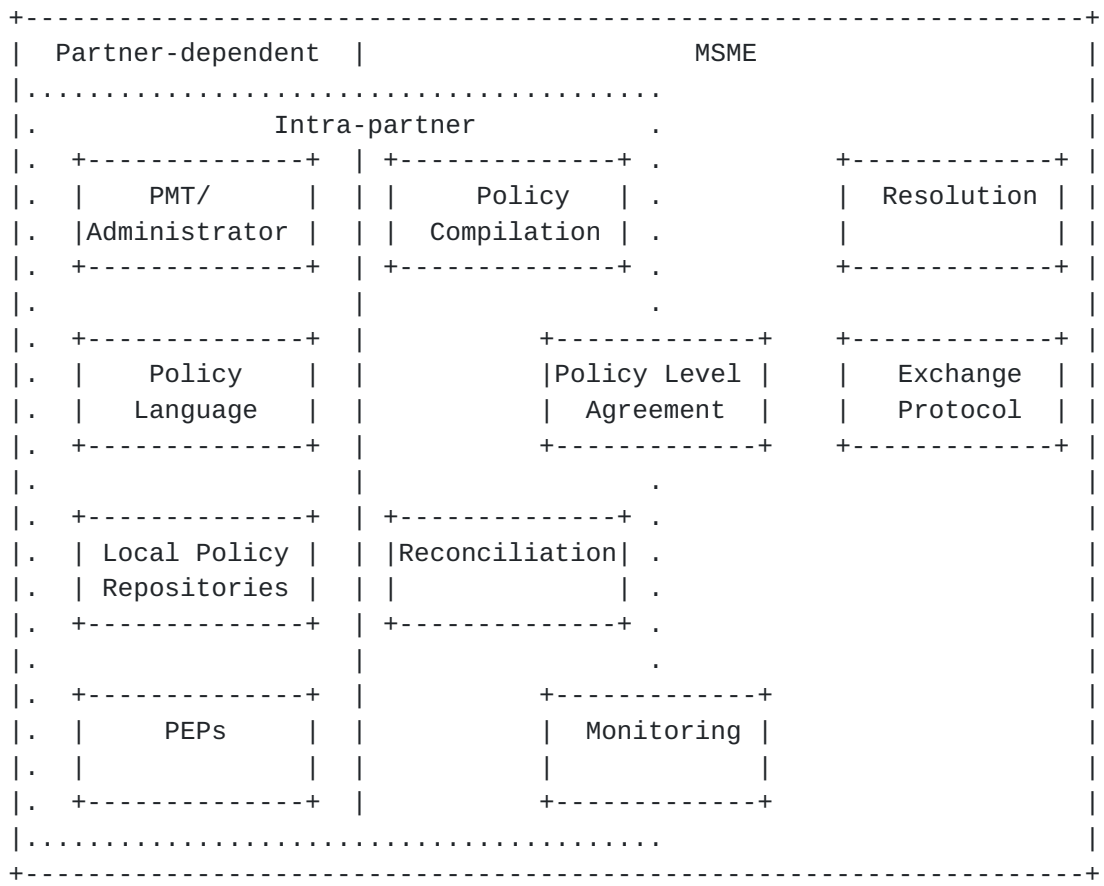


Figure 1. MSME Architecture Components

The remainder of this section will describe the components and their interactions in some detail and will provide a description of data flow through the system to illustrate how the system is designed to work.

2.4.1 MSME Components

2.4.1.1 Policy Level Agreement

Policy level agreements (PLAs) are the means by which each of the partners in a coalition expresses and exchanges its policy. The PLAs are expressed in a common language (PLAL) to facilitate exchange and processing.

PLAs include the following information:

* The identity of the coalition to which a PLA applies
* The members of the coalition
* The member that created the PLA
* Versioning information for the PLA
* Abstract policy rules using arbitrary identifiers chosen by each partner Some identifiers may be agreed upon to be globally unique.
* Bindings that map the identifiers to concrete assets or security services.

Policy rules are defined at an abstract level, with conditions expressed in terms of abstract entities or entity groups and actions expressed in terms of security services and abstract mechanisms. The services and mechanisms are a superset of those described in [iso7498-2].

Each abstract entity, service, or mechanism includes a textual name which identifies one or more bindings which map the abstract part of the rule to a concrete implementation (e.g., an IP address, or a security protocol and its acceptable algorithms). Multiple bindings may have the same name, since the required service may be implemented in multiple security contexts (e.g., IPsec, TLS).

Some binding names may be globally recognized by the coalition partners so that a partner may refer to other partners' entities without having concrete bindings. For example, partner 1 can have an abstract policy that requires ESP to be used between its gateways and partner 2's gateways, without having to know the IP addresses of partner 2's gateways when defining its policy. Those addresses will be supplied by partner 2 in its PLA and will be bound to partner 1's policy during the resolution process.

2.4.1.2 Policy Compilation

Compilation is the process by which each partner generates a PLA from its abstract policy rules. The compilation process takes the set of abstract policy rules and a database of bindings, assembles the bindings needed to support the policy rules, confirms all the bindings are available, other than those supposed to be supplied by other partners, and checks to make sure that the policy rules are

self-consistent (i.e., the policies don't contain any contradictions). The output of the process is either a complete PLA that may be

exchanged with other partners, or errors indicating that the PLA is not consistent or complete and must be fixed by the policy administrator.

2.4.1.3 Resolution

Resolution is the process of combining all (or some subset) of the partners' PLAs into a single PLA containing rules consistent with all of their policies. The 'combined' PLA is known as a resolved PLA (RPLA).

Resolution combines two or more member PLAs in an attempt to generate a single RPLA that is consistent with all of the policies expressed in the individual PLAs. It identifies the security mechanisms supported by all the parties required for each mission-related policy rule. It also identifies the cases where all partners do not support a common security mechanism for a policy rule, an indication that a communication that cannot be successfully initiated due to current policies, thus allowing the policy administrators to address any potential deficiencies in their respective policy specifications. This `diagnostic' use of resolution is an important function in real world applications.

Resolution may be performed in a centralized manner, using a single resolver for the coalition; in a decentralized manner, with each member performing an independent resolution of the same set of PLAs; or in some intermediate way. The MSME resolution mechanism should be designed to support decentralized operations, as centralized operations can be viewed as a special case of the more general decentralized resolution architecture.

2.4.1.4 Exchange Protocol

The PLA exchange protocol is a common protocol is used for the exchange of PLAs and RPLAs, and for monitoring the status of the resolution process.

The exchange of PLAs and RPLAs between all partners requires some exchange mechanism. Although nothing fundamentally precludes partners from implementing differing exchange mechanisms on a bilateral or multilateral basis, it is useful to consider a common exchange protocol. Even if such a protocol is not universally adopted, its architectural framework is helpful in ensuring that whatever exchange mechanisms are implemented sufficiently support the requirements of the MSME system.

The protocol's primary aim is to support the resolution process, by allowing PLAs and RPLAs to be exchanged, and by allowing resolution

status information to be conveyed. The protocol should support the following operations:

* Exchange of PLAs between partners, or from partners to a central resolution authority.

* Exchange of RPLAs: In a centralized resolution architecture, centrally resolved RPLAs will have to be propagated to coalition partners. In a totally decentralized environment, partners may still need to request each others' RPLAs for reconciliation purposes.

* Co-ordination/synchronization of the resolution process: The resolution process is essentially asynchronous, regardless of whether it is performed centrally or in a distributed manner. Some mechanism is required to inform entities within the coalition of the current status of the resolution process. This may, for example, include a means for requesting a new round of resolution, or for of informing partners of the (un-)successful completion of a resolution process.

The protocol must authenticate the source of the PLA and optionally provide data confidentiality for the exchange.

2.4.1.5 Reconciliation

Once an RPLA has been generated, a reconciliation process must occur so that each partner may verify that the RPLA is correct and that it does not conflict with its internal policy rules.

A coalition partner cannot assume that any resolution that it did not perform itself will yield a correct RPLA. The partner must confirm that: no policy rules introduced to the coalition by the partner in its PLA have been removed, no bindings have been defined in a manner that conflicts with any of the definitions introduced by the partner in its PLA, and no policy rules have been introduced in the RPLA that conflict with any of the partner's policy rules in its PLA. Additionally, a partner may use reconcilliation to identify places where resolution occurred correctly, but did not achieve the desired result, and may have to be adjusted by administrative intervention. For example, it could detect policy rules that are not implementable because no common security mechanisms exists with other partners, or cases where a weaker, though still acceptable, mechanism was selected instead of a stronger, more desireable one.

An incorrect RPLA may be generated for a number of reasons, including incorrect implementation of the resolution algorithm, PLA synchronization issues, or malicious intent by one of the coalition partners. In order to ensure the correctness of a new RPLA that it has received, a partner should evaluate its freshness and correctness before provisioning it.

2.4.1.6 Monitoring

Monitoring ensures that RPLAs are being correctly generated and enforced. A number of elements of the MSME system will be involved in the process of monitoring that RPLAs are correctly implemented. The cornerstone of the monitoring aspect of MSME is the reconciliation process, which provides each partner with an essential sanity check that the MSME system is generating correct policy. The versioning mechanism provided by the PLAL provides a means to allow partners to compare the versions of the RPLAs that they are using to ensure that they are operating with consistent policies. Version checking is further facilitated by the MSME PLA exchange protocol which allows partners to query the status of the resolution process, and to obtain the currently active RPLA.

Additional monitoring tasks may include verifying that enforcement points are being configured with the correct policies and that they are using them correctly. This monitoring is necessarily limited since most communications are likely to be encrypted and a partner will generally only have access to the unencrypted messages on its own hosts.

2.4.2 Partner-Dependent Components

2.4.2.1 Policy Management Tool

The policy management tool (PMT) provides a user interface to the MSME system. It provides an administrator the means to enter abstract policies and bindings into the system. The PMT also must be able to read the policies in a PLA or local repository and display the results to the administrator. The administrator should be able to modify these policies.

Policy management goes beyond just creating and viewing policies. The PMT should provide an interface to other functions. For example, it should be able to initiate the policy compilation process and display the results, including any warnings or errors that it produces, so the administrator can correct them. It may also be responsible for initiating resolution, or at least for sending PLAs to a central resolution point.

2.4.2.2 Policy Language

The choice of languages that a partner uses for defining its internal policies are up to the partner, however there must be tools to translate those languages to and from the PLA language in order to create bindings and to translate RPLAs into the language(s) that the local managment system uses to provision policy enforcement points.

2.4.2.3 Local Policy Repositories

Each partner maintains collections of information relating to their existing internal policy environments. The nature of these repositories is highly partner-specific. There may be databases specifically for MSME data, or they may support both MSME and internal policy management systems and provide the link between the two. This section discusses what information the repositories must be able to provide MSME.

The following information should be available:

* Asset data: Mappings between high-level assets and their composition in terms of low-level assets.

* Security service to mechanism mappings: Whenever a high-level policy requires a security service, this repository is consulted to determine what mechanisms are available to implement that service.

* Local mechanism-specific security policy repositories

* Optionally, repositories of global data

The information in the local repositories is used in the generation of bindings. For each asset specified in the high-level policy, the corresponding low-level endpoints are determined from the asset repository. Then for each low-level mechanism, the relevant mechanism-specific repository is consulted to extract the information required for policy resolution.

The local policy repository for each context is consulted to determine which bindings are valid (from local context-specific policy considerations) and may be included in the PLA. This process must also be repeated with the resolved PLAs to ensure that the RPLA does not violate any local policy constraints.

2.4.2.4 Policy Enforcement Points

Once a partner has received and successfully reconciled an RPLA, it is up to the policy enforcement points to enforce the resolved policies. This may require the partner to provision new policy rules to its internal PEPs.

This will typically involve reprovisioning policy management and enforcement agents within the partner's security domain. While this provisioning is outside the scope of the MSME system, any MSME implementation will have to consider the interaction between the RPLA validation and acceptance mechanism, which is part of the core MSME system, and the policy provisioning system, which is outside of it.

The nature of this link between MSME and policy provisioning mechanisms is highly dependent on the nature of the specific provisioning mechanisms in use. Conceptually, it would typically

involve the inverse of the process of extracting information out of the local policy repositories. The new policy can then be distributed by mechanisms such as those being developed by the IPSP working group.

2.4.3 Data Flow

This section describes how the components of the MSME system work together to provide policy management for dynamic coalitions.

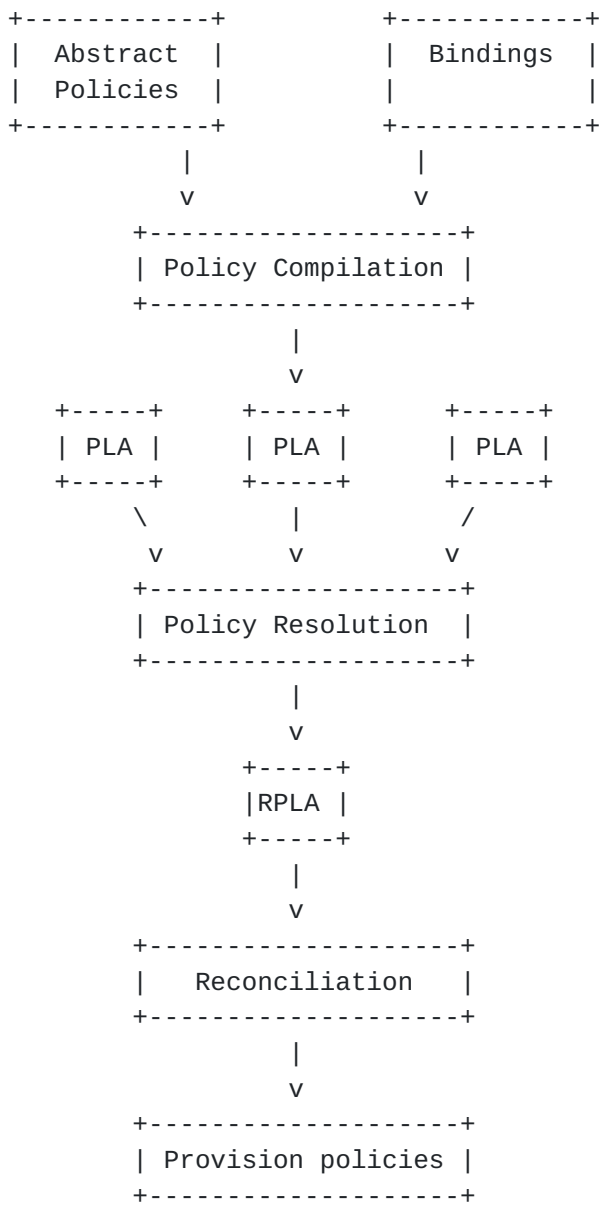


Figure 2: MSME Data Flow

Policies are created by policy administrators. The people who create the abstract policies and the bindings do not have to be the same, since the people who understand the coalition's communication requirements are not necessarily the same as those who understand the concrete policies that the partner supports.

The MSME resolution process may be initiated for several reasons, including a change in policy, abstract or concrete, or a change in the coalition membership.

A partner must first generate a new PLA when a resolution exchange is initiated and a partner determines that its policies (including abstract rules, bindings, and contexts) have changed. If its policies have not changed then its previous PLA may be reused. To generate a new PLA, the compilation process is invoked. It takes the abstract policy rules and the database of bindings, extracts the necessary bindings and produces a consistent PLA, if possible. If there were errors, they must be corrected manually by the policy administrators.

The PLA is sent, using the exchange protocol to one or more policy resolvers that will be creating RPLAs. The policy resolver creates an RPLA that satisfies the requirements of all the RPLAs, if possible. If it is not possible, then an error is returned and the partners need to manually correct their PLAs, if they wish, and attempt the resolution again. Once the RPLA is produced it is sent to the partners, as needed.

The first step a partner performs upon receiving a new RPLA is reconciliation to ensure that the RPLA is in fact consistent with its policies. If reconciliation fails, the partner should correct the problem, if possible, or report it to the other partners which may have to make corrections to their policies to correct the problems.

If reconciliation succeeds, the partner should commence enforcement of the RPLA. Again, the details of the operations required to enforce the RPLA are partner-specific, but the overall principles are the same: new concrete rules may need to be provisioned at enforcement points to accommodate coalition policy. This necessarily implies a narrowing of existing policy, since a broadening would require human negotiation (if resolution failed, for example, one or more partners might agree to relax their local policy restrictions to allow resolution to succeed).

3. Ideas for IPSP

The MSME system is attempting to solve a different problem than is currently being addressed by the IPSP working group so while its architecture contains many similar elements, they are instantiated differently. However, since it is also working in the security policy management domain it may provide some insights that can improve the final output of the IPSP WG, especially in the area of defining and generalizing policy rules. This section explores some of the areas where MSME's work may provide some insights into improving the IPSP work.

3.1 Late Name Binding

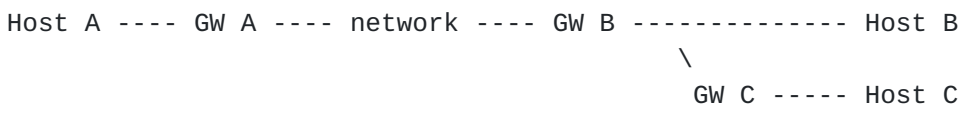
Early IPSP working group drafts, now expired, such as the "Security Policy Specification Language" (draft-ietf-ipsp-spsl-00.txt) and "Security Policy Protocol" (draft-ietf-ipsp-spp-00.txt) included the

beginnings of an idea that some hosts should not be specified by IP address, but could be addressed in a more general way. This allowed policies, which were configured in the security servers, to be defined in such a way that it did not need to specify an IP address for hosts or gateways it may not know the IP address, or even if such a gateway might exist. For example, those two drafts allowed policies to use some defined strings such as "host", "remote-sg", or "local-sg" to refer to such entities.

While it was an interesting idea, it was poorly developed and, as described in those drafts, not completely implementable. While the "host" tag could easily be interpreted as either the source or destination of the communication, none of the others could be reliably translated into a specific host or gateway. MSME provides some insights into how to design this feature in an implementable fashion and possibly ways to expand on the idea. In particular adding the idea of binding a name to one or more assets could add this flexibility.

The protocol and language could continue to use standardized strings to refer to particular types of hosts as those drafts indicated. However, the security servers would use bindings as part of their local policies to map the strings to the appropriate network entities. Their mapping could even be different for different policy rules so that a communication being negotiated for host A could have a different mapping than a communication being negotiate for host B, for example.

To illustrate this, let's look at the following example:



Host A initiates a request to discover the policy requirements for a communication between itself and Host B. GW A has a policy that requires a tunnel for that communication between itself and any gateway that is authoritative over Hosts B or C (indicated by the string "remote-sg"). GW B (or its policy server) has a binding that maps the string "remote-sg" to GW B for this communication so GW A then learns as part of the resolution that it needs to have a tunnel with GW B.

If Host A then wants to communicate with Host C and the policies are the same, GW B can have a binding that maps "remote-sg" to GWs B and C so GW A can tunnel to either of them.

We can extend this concept beyond a few standardized strings that all policy domains can implement. If we allow arbitrary names to be used (or at least a private namespace), then security policy domains can

privately agree upon names to use to abstract portions of their policy domains.

For example, if companies X and Y need to facilitate communications between their engineering departments, they could agree upon "x_eng"

to be a name for X's engineering department and "y_eng" for Y's. X can have a binding that maps "x_eng" to the necessary hosts in its engineering department and Y can do similarly. Now X can create policies in terms of "y_eng" instead of a list of all the IP addresses in Y's engineering department.

This has a couple of advantages. Now the hosts in Y's engineering department can change without Y having to inform X, since all Y needs to do is change its binding and X's policy is still valid since the name does not change. Additionally, it can help reduce the amount of network information that each company needs to expose to the other.

While it may be possible to achieve some of this abstraction by using wildcarded DNS names (e.g., *.eng.y.com) in policy rules, the binding mechanism is much more powerful and general. Not only may it map from a name to one or more hosts, but it can map from a name to one or more conditions that are required (e.g., port, protocol, etc). For example, company Y could have different bindings for its HTTP and FTP servers, even if they are the same hosts, with the same host names, since each binding would map the binding name to a set of hosts and a port number.

3.2 Generalization

Currently, IPSP is looking exclusively at providing security policy management for the IPsec security domain. If this can be accomplished in a more general manner that allows the solution to be applied to other security domains, it would increase the usefulness of the work, without greatly increasing its scope.

One possible means of generalizing the IPSP work is to allow policy rules to express multiple security domain options in one policy rule by using bindings.

In order to accomplish this, policy rules must be able to be expressed using names for the source and destination of the condition and the action. For example, Company Y could have the policy:

```
src y_eng dst x_eng -> strong_authentication
```

Now, suppose both companies X and Y are willing to use either IPsec or TLS to protect the communication. "y_eng," "x_eng," and "strong_authentication" can refer to bindings that have both IPsec and TLS mappings which can be sent along with the policy rule as part of the policy discovery phase of IPSP's protocol. "y_eng" may map to a set of IP addresses and port numbers in the IPsec context. In the TLS context it may map to a set of user names. "strong_authentication" may also map to the required mechanisms in each context such as an ESP proposal or a TLS mac algorithm.

Company X can then have a similar policy:

```
   src y_eng dst x_eng -> xs_authentication
```

X will have mappings for "x_eng" and "xs_authentication" in a similar manor to Y. Note that "x_eng" and "y_eng" can be pre-agreed upon names between the two companies as described above. "strong_authentication" and "xs_authentication" are not, however.

Resolution of the two policies is performed at the mechanism level, as would be done if the bindings did not exist. Only bindings in the same context can be resolved and each context is resolved independently. This allows the resolution to succeed if at least one condition/action pair in one context can successfully resolve. If multiple contexts successfully resolve, then both may be returned as the answer.

4. Acknowledgments

The author thanks the MSME team, Luis Sanchez, Charlie Lynn, Geva Patz, Alex Colvin, David Waitzman, and Rajesh Krishnan for their work developing the MSME system and reviewing this draft.

5. References

[Bra97] Bradner, S., "Key Words for use in RFCs to indicate Requirement Levels", RFC2119, March 1997.

[TERM] A. Westerinen, J. Schnizlein, J. Strassner, M. Scherling, B. Quinn, S. Herzog, A. Huynh, M. Carlson, J. Perry, and S. Waldbusser, Terminology for Policy-Based Management draft-ietf-policy-terminology-04.txt (approved for RFC)

[iso7498-2] ISO. "Information Processing Systems - Open Systems Interconnection - Basic Reference Model - part 2: Security Architecture," first edition, International Standard ISO 7298-2, ISP, February 1989.

Disclaimer

The views and specification here are those of the authors and are not necessarily those of their employers. The authors and their employers specifically disclaim responsibility for any problems arising from correct or incorrect implementation or use of this specification.

Author Information

Matthew N. Condell
BBN Technologies
10 Moulton Street
Cambridge, MA 02138
USA
Email: mcondell@bbn.com
Telephone: +1 (617) 873-6203