IPSP Working Group
Internet Draft

A. Keromytis
U. of Pennsylvania
M. Richardson
Sandelman Software Works
L. Sanchez
BBN/GTEI
October 1999

draft-keromytis-ipsp-arch-00.txt

IPsec Policy Discovery Architecture

Status of this Memo

Abstract

This document describes an IP Security Policy architecture that conforms to the requirements set forth in [IPSP-REQ]. The architecture defines the mechanisms and protocols needed for discovering, accessing, and processing security policy information of varying granularity. The architecture accommodates topology and policy changes without need of manual reconfiguration of clients and security gateways.

1. Introduction

The Security Policy System (SPS) defines a distributed database of security policy information. It provides the mechanisms needed for discovering, accessing, and processing security policy information of hosts, subnets, or networks.

In the SPS architecture there are two types of systems, Policy Servers and Policy Clients. Policy Servers provide a security policy repository that Policy Clients (end hosts and security gateways) may consult to determine what the security parameters for

a particular communication should be.

Policy Clients must be configured to know their Policy Server(s). This configuration may be manual or through some automated discovery mechanism.

Policy Servers must be provided with the security policy for the systems it is responsible for. How the policy is determined and stored in the Policy Server itself is outside the scope of this specification.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119].

1.1 Terminology

See [IPSP-REQ] for the terminology used throughout this document.

2. Architecture Overview

A host (H1) that wishes to communicate with a remote host (H2) first needs to decide what the end-to-end security parameters of that communication should be. These may be stored or otherwise determined locally (e.g., an application required some specific form of traffic protection, through some OS-specific mechanism), or may be retrieved from its appointed Policy Server. The Policy Server may also know exactly what SAs need to be negotiated, either because it was configured with that information or because it has cached the result of a previous lookup.

If the Policy Server does not have such cached information, the host enters a discovery phase, wherein it tries to establish what security gateways lie in the path to the remote endpoint. This is achieved through a special discovery message that is sent to the remote endpoint.

A security gateway that intercepts this message consults with its Policy Server to determine what security parameters are required for the packets from H1 to H2. The Policy Server may have cached information from previous runs of the discovery process, in which case it is returned to the security gateway (and from there, to the origin of the discovery process). If not, the security gateway forwards the discovery message to H2, along with its security requirements for packets from H2 to H1 (the reverse path). Each subsequent security gateway that intercepts this message repeats this step. If at any point a Policy Server does have cached information relevant to this discovery process, it may provide it and cause the discovery process to end.

Eventually, the discovery message reaches the Security Gateway

"closest" to host H2, which consults its Policy Server. If the Policy Server has not been configured to provide Security Policy information for host H2, the Security Gateway will forward the discovery message to host H2. In that case, host H2 determines the security policy for the end-to-end communication (possibly by querying its Policy Server). H2 then sends a message to that last security gateway, describing the security requirements for packets from H1 to H2.

That security gateway adds to these requirements its own (that is, the requirements for packets from H1 to that security gateway). This step is repeated until H1 receives a message from its closest security gateway that contains the security requirements of all the security gateways along the path to H2, as well as H2's requirements. At the end of this process, both H1 and H2 have a list of security requirements of all the security gateways along the path between them. If H2 is not satisfied with these requirements, it may initiate a discovery process to H1 on its own.

Furthermore, at each step where a host or security gateway receives policy information, it MAY forward such information to its Policy Server for caching, and it may cache the policy information locally.

3. Security Gateway Discovery Protocol

This section gives a brief overview of the Security Gateway Discovery Protocol. A separate document will describe the protocol in more detail.

The Security Gateway Discovery Protocol (SGDP) is implemented as a transport protocol over IP.

The SGDP allows the initiator of the discovery process to determine the security policy of all Security Gateways between itself and the remote end-host, as well as the security policy of the end-host itself. The origin end-host or Security Gateway would then establish the necessary SAs (or reuse existing SAs where applicable/feasible) with the Security Gateways and remote end-host, using some automated SA-negotiation protocol, such as IKE [RFC-2409] or Photuris [Photuris].

The security policy information acquired through the discovery process includes parameters for the SAs that should be established with a particular SG or end-host, security credentials (e.g., certificates) that should be used by the SA-negotiation protocol, as well as other (as yet undefined) information.

To compensate for topology changes, the discovery process may be instructed to discard cached information, forcing a complete path

traversal to occur.

4. Security Policy Server Operation

The following sections discuss the operational requirements of a Policy Server in the IPSP architecture.

4.1 Security Policy Server Query Protocol

This section gives a brief overview of the Security Policy Server Query Protocol (SPSQP). A separate document will describe the protocol in more detail.

The SPSQP supports the following primitives:

* Client registration (and de-registration).

* Client download of their security policy.

* Client download of cached security policy.

* Client upload of their security policy.

* Client upload of security policy found in a received SGDP Discovery Message for caching purposes.

* Server push of (updated) security policy to registered clients.

A Policy Server MUST verify the legitimacy of clients downloading or uploading security policy information. Clients MUST verify the identity of the Policy Server.

All communications over SPSQP MUST be secured. IPsec MUST be used to secure all communications between Policy Servers and clients (end-hosts and Security Gateways), and verify the legitimacy of Policy Servers and clients.

4.2 Security Policy Caching

A Policy Server, SG, or end-host MAY choose to cache policy (and policy-related, such as credentials) information acquired through the discovery process. Policy information MUST have an indication as to what its cache lifetime is. Policy Servers, SGs, and end-hosts that cache policy information MUST remove expired policy entries from their caches.

Note that some policy objects MAY specify other types of expiration mechanisms (e.g., online validity checks). There are two classes of such objects:

* Policy information that is verified by SGs and end-hosts (e.g.,

certificates and other types of credentials). Policy Servers, SGs, and end-hosts MAY periodically verify the validity of such information in their caches. Invalid entries MUST be discarded.

* Policy information that is used-as is by SGs and end-hosts. Policy Servers, SGs, and end-hosts that cache such objects MUST periodically verify the validity of such information in their caches. Invalid entries MUST be discarded.

4.3 Security Policy Decorrelation

Security Policies MUST be decorrelated. More text will be added in this section, explaining the necessity and mechanics of decorrelation. A separate document will describe the decorrelation algorithm in more detail.

4.4 Policy Server Discovery

An end-host or Security Gateway MUST be configured so that it can contact its Policy Server. Such information may include network addresses, security credentials (for IPsec), etc.

Some automated mechanism for discovering a Policy Server MAY be defined as part of this architecture. It is not yet known what form this mechanism may take.

4.5 Where to Place a Policy Server

A Policy Server may co-exist with a Security Gateway, an end-host, or may reside on a system by itself.

More text is needed here, explaining the tradeoffs and issues involved with the different placements.

5. End-host Requirements

An end-host that complies with the IPSP architecture MUST be able to initiate a Security Gateway discovery process (see Section 3). If the end-host is expected to operate inside a Security Domain, it MUST implement the client side of the Security Policy Server Query Protocol (see Section 4.1). Implementation of the server side of the SPSQ protocol is optional.

6. Legacy End-hosts

This section describes IPSP operation when either or both of the end-hosts (origin and/or destination end-hosts) are not IPSP-aware.

6.1 Legacy Origin End-host

When an origin end-host operating inside a Security Domain does not implement the Security Gateway Discovery Protocol, coordination between Security Gateways and the end-host is not possible.

A Security Gateway that intercepts a packet from such a host MAY initiate a Security Gateway discovery process, specifying that it will be proxying traffic for the end-host. This will allow the Security Gateway to establish IPsec tunnels with other Security Gateways (and potentially the destination end-host itself) that protects the origin end-host's traffic. The "reverse channel" policy added to the discovery packet assumes that the destination end-host implements SGDP (and can thus take advantage of this information). If that is not true, a Security Gateway discovery process will have to be initiated in the reverse direction by the last Security Gateway (see section 6.2).

6.2 Legacy Destination End-host

When a destination end-host does not implement the SGDP, it is the responsibility of the Policy Server of its Security Domain to specify the end-to-end security parameters (if any). This means that a Policy Server MUST be aware of which hosts it is responsible for.

The Policy Server responsible for a legacy destination end-host MAY initiate a Security Gateway discovery process in the reverse direction if the origin end-host does not implement SGDP (i.e., the discovery process was a proxy one). However, some mechanism needs to be employed to avoid an endless loop of Security Gateway discovery.

7. Legacy Security Gateways

Legacy Security Gateways do not participate in the discovery process, since they do not implement the SGDP. Such a system, upon receipt of a discovery packet may drop it (which will cause the discovery process to time-out), forward it with no further processing, or initiate an IPsec exchange with some remote host or Security Gateway, based on its local (non-IPSP-conforming) security policy. In the latter two cases, no further action is required by any IPSP-compliant system, as the legacy Security Gateway is transparent to the discovery process.

If the legacy Security Gateway drops the discovery packets and sends back an appropriate ICMP message, the recipient of such a message (another SG or the origin end-host) MAY establish the necessary IPsec SAs with the legacy SG to allow traffic to flow through the legacy SG. The legitimacy of the ICMP message MUST be verified through cryptographic (or other) means.

Alternatively, the Security Gateway or origin end-host MUST terminate the discovery process and notify the Policy Servers, SGs, and origin end-host involved in the discovery process.

No solution as yet exists if the legacy Security Gateway silently discards packets.

8. Security Considerations

This section has not been completed. It will be in future versions of this draft.

9. IANA Considerations

A new transport protocol number for the SGDP needs to be assigned by IANA. No further actions by IANA are required (yet).

References:

[RFC-2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC-2119, March 1997.

[RFC-2401] S. Kent, R. Atkinson, RFC2401: "Security Architecture for the Internet Protocol", November 1998.

[IPSP-REQ] M. Richardson, A. Keromytis, L. Sanchez, draft-ietf-ipsp-requirements-00.txt: "IPsec Policy Discovery Protocol Requirements", October 1999

[RFC-2409] Harkins, D., and D. Carrel, "The Internet Key Exchange (IKE)", RFC 2409, November 1998. Authors' Address

[Photuris] Karn, P., and B. Simpson, Photuris: Session Key Management Protocol, Work in Progress.

Authors' addresses:

Angelos D. Keromytis
Distributed Systems Lab
CIS Department, University of Pennsylvania
200 S. 33rd Street
Philadelphia, Pennsylvania 19104-6389

Telephone: +1 215 573 3639
Email: angelos@dsl.cis.upenn.edu

Michael C. Richardson
Sandelman Software Works Corp.

152 Rochester Street
Ottawa, ON K1R 7M4
Canada

Telephone: +1 613 276-6809
Email: mcr@sandelman.ottawa.on.ca

Luis A. Sanchez
BBN Technologies
GTE Internetworking
10 Moulton Street
Cambridge, MA 02140
USA

Telephone: +1 (617) 873-3351
Email: lsanchez@bbn.com

Expiration and File Name

This draft expires April 1, 2000

Its file name is draft-keromytis-ipsp-arch-00.txt