Network Working Group
Request for Comments: 2207
Category: Standards Track

L. Berger
FORE Systems
T. O'Malley
BBN

September 1997

RSVP Extensions for IPSEC Data Flows

Status of this Memo

Abstract

This document presents extensions to Version 1 of RSVP. These extensions permit support of individual data flows using RFC 1826, IP Authentication Header (AH) or RFC 1827, IP Encapsulating Security Payload (ESP). RSVP Version 1 as currently specified can support the IPSEC protocols, but only on a per address, per protocol basis not on a per flow basis. The presented extensions can be used with both IPv4 and IPv6.

Table of Contents

1 Introduction

Recently published Standards Track RFCs specify protocol mechanisms to provide IP level security. These IP Security, or IPSEC, protocols support packet level authentication, [RFC 1826], and integrity and confidentiality [RFC 1827]. A number of interoperable implementations already exist and several vendors have announced commercial products that will use these mechanisms.

The IPSEC protocols provide service by adding a new header between a packet's IP header and the transport (e.g. UDP) protocol header. The two security headers are the Authentication Header (AH), for authentication, and the Encapsulating Security Payload (ESP), for integrity and confidentiality.

RSVP is being developed as a resource reservation (dynamic QoS setup) protocol. RSVP as currently specified [RFC 2205] is tailored towards IP packets carrying protocols that have TCP or UDP-like ports. Protocols that do not have such UDP/TCP-like ports, such as the IPSEC protocols, can be supported, but only with limitations. Specifically, for flows of IPSEC data packets, flow definition can only be done on per IP address, per protocol basis.

This memo proposes extensions to RSVP so that data flows containing IPSEC protocols can be controlled at a granularity similar to what is already specified for UDP and TCP. The proposed extensions can be used with both IPv4 and IPv6. Section 2 of this memo will provide an overview of extensions. Section 3 contains a description of extended protocol mechanisms. Section 4 presents extended protocol processing rules. Section 5 defines the additional RSVP data objects.

2 Overview of Extensions

The basic notion is to extend RSVP to use the IPSEC Security Parameter Index, or SPI, in place of the UDP/TCP-like ports. This will require a new FILTER_SPEC object, which will contain the IPSEC SPI, and a new SESSION object.

While SPIs are allocated based on destination address, they will typically be associated with a particular sender. As a result, two senders to the same unicast destination will usually have different SPIs. In order to support the control of multiple independent flows between source and destination IP addresses, the SPI will be included as part of the FILTER_SPEC. When using WF, however, all flows to the same IP destination address using the same IP protocol ID will share the same reservation. (This limitation exists because the IPSEC transport headers do not contain a destination demultiplexing value like the UDP/TCP destination port.)

Although the RESV message format will not change, RESV processing will require modification. Processing of the new IPSEC FILTER_SPEC will depend on the use of the new SESSION object and on the protocol ID contained in the session definition. When the new FILTER_SPEC object is used, the complete four bytes of the SPI will need to be extracted from the FILTER_SPEC for use by the packet classifier. The location of the SPI in the transport header of the IPSEC packets is dependent on the protocol ID field.

The extension will also require a change to PATH processing, specifically in the usage of the port field in a session definition. An RSVP session is defined by the triple: (DestAddress, protocol ID, DstPort). [RFC 2205] includes the definition of one type of SESSION object, it contains UDP/TCP destination ports, specifically "a 16-bit quantity carried at the octet offset +2 in the transport header" or zero for protocols that lack such a field. The IPSEC protocols do

not contain such a field, but there remains a requirement for demultiplexing sessions beyond the IP destination address. In order to satisfy this requirement, a virtual destination port, or vDstPort, is introduced. The vDstPort value will be carried in the new SESSION object but not in the IPSEC transport header. The vDstPort allows for the differentiation of multiple IPSEC sessions destined to the same IP address. See Section 5 for a discussion of vDstPort ranges.

In PATH messages, the SENDER_TEMPLATE for IPSEC flows will have the same format as the modified FILTER_SPEC. But, a new SESSION object will be used to unambiguously distinguish the use of a virtual destination port.

Traffic will be mapped (classified) to reservations based on SPIs in FILTER_SPECs. This, of course, means that when WF is used all flows to the same IP destination address and with the same IP protocol ID will share the same reservation.

The advantages to the described approach are that no changes to RFC1826 and 1827 are required and that there is no additional per data packet overhead. Appendix A contains a discussion of the advantages of this approach compared to several other alternatives. This approach does not take advantage of the IPv6 Flow Label field, so greater efficiency may be possible for IPv6 flows. The details of IPv6 Flow Label usage is left for the future.

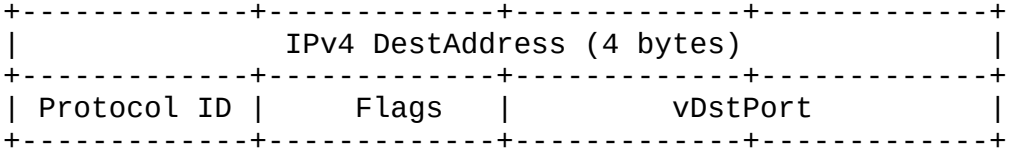
3 Object Definition

The FILTER_SPEC and SENDER_TEMPLATE used with IPSEC protocols will contain a four byte field that will be used to carry the SPI. Rather than label the modified field with an IPSEC specific label, SPI, the label "Generalized Port Identifier", or GPI, will be so that these object may be reused for non-IPSEC uses in the future. The name for these objects are the IPv4/GPI FILTER_SPEC, IPv6/GPI FILTER_SPEC, IPv4/GPI SENDER_TEMPLATE, and IPv6/GPI SENDER_TEMPLATE. Similarly, the new SESSION objects will be the IPv4/GPI SESSION and the IPv6/GPI SESSION. When referring to the new objects, IP version will not be included unless a specific distinction between IPv4 and IPv6 is being made.

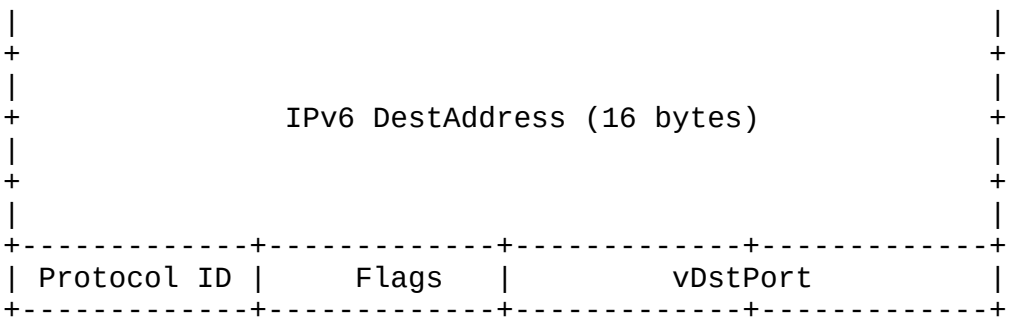
3.1 SESSION Class

SESSION Class = 1.

o IPv4/GPI SESSION object: Class = 1, C-Type = 3



o IPv6/GPI SESSION object: Class = 1, C-Type = 4

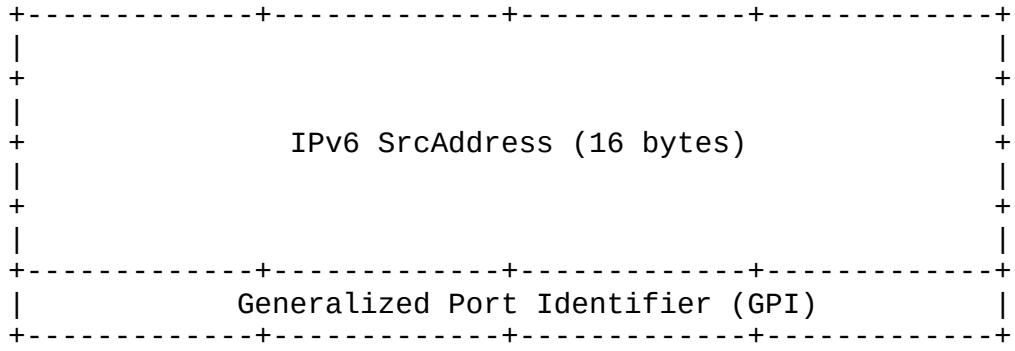


3.2 FILTER_SPEC Class

FILTER_SPEC class = 10.

o IPv4/GPI FILTER_SPEC object: Class = 10, C-Type = 4

o IPv6/GPI FILTER_SPEC object: Class = 10, C-Type = 5



3.3 SENDER_TEMPLATE Class

SENDER_TEMPLATE class = 11.

o IPv4/GPI SENDER_TEMPLATE object: Class = 11, C-Type = 4

Definition same as IPv4/GPI FILTER_SPEC object.

o IPv6/GPI SENDER_TEMPLATE object: Class = 11, C-Type = 5

Definition same as IPv6/GPI FILTER_SPEC object.

4 Processing Rules

This section presents additions to the Processing Rules presented in [RFC 2209]. These additions are required in order to properly process the GPI SESSION and FILTER_SPEC objects. Values for referenced error codes can be found in [RFC 2205]. As in with the other RSVP documents, values for internally reported (API) errors are not defined.

4.1 Required Changes

Both RESV and PATH processing will need to be changed to support the new objects. The changes ensure consistency and extend port processing.

The following PATH message processing changes are required:

- When a session is defined using the GPI SESSION object, only the GPI SENDER_TEMPLATE may be used. When this condition is violated, end-stations should report a "Conflicting C-Type" API error to the application.

- For PATH messages that contain the GPI SESSION object, end-stations must verify that the protocol ID corresponds to a protocol known to use the GPI SESSION object. Values 51 (AH) or 50 (ESP) must be supported by implementations supporting the described IPSEC extensions. If an unknown protocol ID is used, then the API should report an "API Error" to the application.

- For such messages, the vDstPort value should be recorded. The vDstPort value forms part of the recorded state and is used to match Resv messages, but it is not passed to traffic control. Non-zero values of vDstPort are required. This requirement ensures that a non-GPI SESSION object will never merge with a GPI SESSION object. Violation of this condition causes an "Invalid Destination Port" API error.

The changes to RESV message processing are:

- When a RESV message contains a GPI FILTER_SPEC, the session must be defined using the GPI SESSION object. Otherwise, this is a message formatting error.

- The GPI contained in the FILTER_SPEC must match the GPI contained in the SENDER_TEMPLATE. Otherwise, a "No sender information for this Resv message" error is generated.

- When the GPI FILTER_SPEC is used, each node must create a data classifier for the flow described by the quadruple: (DestAddress, protocol ID, SrcAddress, GPI). The data classifier will need to look for the four byte GPI at transport header offset +4 for AH, and at transport header offset +0 for ESP.

4.2 Merging Flowspecs

When using this extension for IPSEC data flows, RSVP sessions are defined by the triple: (DestAddress, protocol Id, vDstPort). Similarly, a sender is defined by the tuple: (SrcAddress, GPI), where the GPI field will be a four byte representation of a generalized source port. These extensions have some ramifications depending upon the reservation style.

4.2.1 FF and SE Styles

In the FF and SE Styles, the FILTER_SPEC object contains the (SrcAddress, GPI) pair. This allows the receiver to uniquely identify senders based on both elements of the pair. When merging explicit sender descriptors, the senders may only be considered identical when both elements are identical.

4.2.2 WF Styles

These extensions provide very limited service when used with WF style reservations. As described, the SENDER_TEMPLATE and FILTER_SPEC each contain the GPI. In a WF style reservation, the RESV message does NOT contain a FILTER_SPEC (after all, it is a wildcard filter), and the SENDER_TEMPLATE is ignored (again, because any sender is allowed). As a result, classifiers may match all packets which contain both the session's destination IP address and protocol ID to such WF reservations.

Although a solution for this limitation is not proposed, this issue is not seen as significant since IPSEC applications are less likely to use WF style reservations.

5 IANA Considerations

The range of possible vDstPort values is broken down into sections, in a fashion similar to the UDP/TCP port ranges.

0	Illegal Value
1 - 10	Reserved. Contact authors.
11 - 8191	Assigned by IANA
8192 - 65535	Dynamic

IANA is directed to assign the well-known vDstPorts using the following criteria: Anyone who asks for an assigned vDstPort must provide a) a Point of Contact, b) a brief description of intended use, and c) a short name to be associated with the assignment (e.g. "ftp").

6 Security Considerations

The same considerations stated in [RFC 2205], [RFC 1826], and [RFC 1827] apply to the extensions described in this note. There are two additional issue related to these extensions.

First, the vDstPort mechanism represents another data element about the IP Flow that might be available to an adversary. Such data might be useful to an adversary engaging in traffic analysis by monitoring not only the data packets of the IP Flow but also the RSVP control messages associated with that Flow. Protection against traffic analysis attacks is outside the scope of this mechanism. One possible approach to precluding such attacks would be deployment and use of appropriate link-layer confidentiality mechansisms, such as encryption.

Secondly, Changes in SPI values for a given flow will affect RSVP flows and reservations. Changes will happen whenever that flow changes its Security Association. Such changes will occur when a flow is rekeyed (i.e. to use a new key). Rekeying intervals are typically set based on traffic levels, key size, threat environment, and crypto algorithm in use. When an SPI change occurs it will, in most cases, be necessary to update (send) the corresponding SENDER_TEMPLATEs and FILTER_SPECs. IPSEC implementations, RSVP applications, and RSVP end-station implementations will need to take the possibility of changes of SPI into account to ensure proper reservation behavior. This issue is likely to be a tolerable, since rekeying intervals are under the control of local administrators.

Many, if not most, RSVP sessions will not need to deal with this rekeying issue. For those applications that do need to deal with changes of SPIs during a session, the impact of sending new PATH and RESV messages will vary based on the reservation style being used. Builders of such applications may want to select reservation style based on interaction with SPI changes.

The least impact of an SPI change will be to WF style reservations. For such reservations, a new SENDER_TEMPLATE will need to be sent, but no new RESV is required. For SE style reservations, both a new SENDER_TEMPLATE and a new RESV will need to be sent. This will result in changes to state, but should not affect data packet delivery or actual resource allocation in any way. The FF style will be impacted the most. Like with SE, both PATH and RESV messages will need to be sent. But, since FF style reservations result in sender receiving its own resource allocation, resources will be allocated twice for a period of time. Or, even worse, there won't be enough resources to support the new flow without first freeing the old flow.

A way around this FF/SPI-change problem does exist. Applications that want FF style reservations can use multiple SE reservations. Each real sender would have a separate SESSION (vDstPort) definition. When it came time to switch SPIs, a shared reservation could be made for the new SPI while the old SPI was still active. Once the new SPI was in use, the old reservation could be torn down. This is less than optimal, but will provide uninterrupted service for a set of applications.

7 References

[RFC 2205] Braden, R., Ed., Zhang, L., Estrin, D., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, September 1997.

[RFC 2209] Braden, R., Ed., Zhang, "Resource ReSerVation Protocol (RSVP) -- Version 1 Message Processing Rules", RFC 2209, September 1997.

[RFC 1825] Atkinson, R., "Security Architecture for the Internet Protocol", RFC 1825, NRL, August 1995.

[RFC 1826] Atkinson, R., "IP Authentication Header", RFC 1826, NRL, August 1995.

[RFC 1827] Atkinson, R., "IP Encapsulating Security Payload", RFC 1827, NRL, August 1995.

8 Acknowledgments

This note includes ideas originated and reviewed by a number of individuals who did not participate in this note's writing. The authors would like to acknowledge their contribution. We thank Ran Atkinson <rja@cisco.com>, Fred Baker <fred@cisco.com>, Greg Troxel <gdt@bbn.com>, John Krawczyk <jkrawczyk@BayNetworks.com> for much appreciated input and feedback. Special appreciation goes to Bob Braden <braden@isi.edu> for his detailed editorial and technical comments. We also thank Buz Owen, Claudio Topolcic, Andy Veitch, and Luis Sanchez for their help in coming up with the proposed approach. If any brain-damage exists in this note, it originated solely from the authors.

9 Authors' Addresses

Lou Berger
FORE Systems
6905 Rockledge Drive
Suite 800
Bethesda, MD 20817

Phone: 301-571-2534
EMail: lberger@fore.com

Tim O'Malley
BBN Corporation
10 Moulton Street
Cambridge, MA 02138

Phone: 617-873-3076
EMail: timo@bbn.com

A Options Considered

This sections reviews other approaches that were explored in developing the described extensions. They are included here to provide additional context into the general problem. All listed options were rejected by the working group.

Four other options were considered:

1. UDP Encapsulation
   Add a UDP header between the IP and the IPSEC AH or ESP headers.

2. FlowID Header Encapsulation
   Add a new type of header between the IP and the IPSEC AH or ESP headers.

3. IPSEC modification
   Modify IPSEC headers so that there are appropriate fields in same location as UDP and TCP ports.

4. AH Transparency
   Skip over the Authentication Header packet classifier processing.

A.1 UDP Encapsulation

Since current SESSION and FILTER object expect UDP or TCP ports, this proposal says let's just give it to them. The basic concept is to add a UDP port between the IP and AH/ESP headers. The UDP ports would provide the granularity of control that is need to associate specific flows with reservations.

Source and destination ports would be used, as normal, in RSVP session definition and control. The port fields would also need to be used to identify the real transport level protocol (e.g. ESP) being used. Also since many UDP ports are assigned as well known ports, use of port numbers would be limited. So, the port fields would need to be used to unambiguously identify 1) the next level protocol, 2) the RSVP session, and 3) the RSVP reservation.

The advantages of this option is that no RSVP changes are required. The disadvantages is that, since the headers aren't in the expected location, RFC 1826 and RFC 1827 are violated.

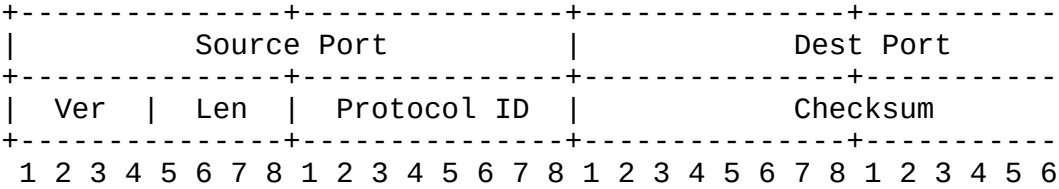
A.2 FlowID Header Encapsulation

[This option was originally proposed by Greg Troxel <gdt@bbn.com>.]

This option is very similar to option 1, but is more generic and could be adopted as a standard solution. The notion is to use UDP like ports for the sole purpose of flow identification. RSVP would treat this new protocol exactly the same as UDP.

The difference between this and UDP encapsulation is in destination host processing. The destination host would essentially ignore port information and use a new field, protocol ID, to identify which protocol should process the packet next. Some examples of protocol IDs correspond to TCP, UDP, ESP, or AH.

The format of the FlowID Header would be:



2 bytes source port
2 bytes dest port
4 bits version (1)
4 bits length-32 (2)
8 bits protocol ID
16 bits checksum

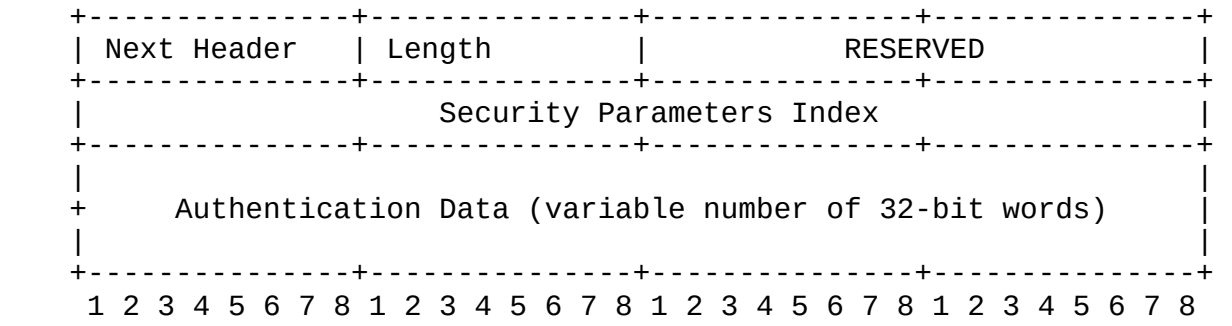
The advantage of this protocol is that flow identification is separated from all other protocol processing. The disadvantage is that the addition of a header violates RFC 1826 and 1827, and also that applications using RSVP will need to add this extra header on all data packets whose transport headers do not have UDP/TCP like ports.

A.3 IPSEC Protocol Modification

The basic notion of this option is to leave RSVP as currently specified and use the Security Association Identifier (SPI) found in the IPSEC headers for flow identification. There are two issues with using the SPI. The first is that the SPI is located in the wrong location when using Authentication (AH). The second issue is how to make use of the SPI.

The first issue is easy to fix, but violates RFC 1826. UDP and TCP have port assignments in the first 4 bytes of their headers, each is two bytes long, source comes first, then destination. The ESP header has the SPI in the same location as UDP/TCP ports, the AH doesn't.

The IP Authentication Header has the following syntax:



Simply reversing the first 4 bytes with the SPI we will have the SPI in the location that RSVP expects. This would be non-standard, or require a major (i.e. not backward compatible) change to RSVP 1826.

The second issue is how to make use of the SPI. Per the current RSVP specification, the first two bytes of a flow's SPI will need to be carried in the PATH message and the second two bytes in the RESV message. The biggest problem is that the SPI is normally selected by the receiver and is likely to be different for EACH sender. (There is a special case where the same SPI is used by all senders in a multicast group. But this is a special case.) It is possible to have the SPI selected prior to starting the RSVPsession. This will work for unicast and the special multicast case. But using this approach means that setup time will usually be extended by at least 1 round trip time. Its not clear how to support SE and WF style reservations.

The advantage of this approach is no change to RSVP. The disadvantages are modification to RFC1827 and limited support of RSVP reservation styles.

A.4 AH Transparency

The source of the RSVP support of IPSEC protocols problem is that the real transport header is not in the expected location. With ESP packets, the real source and destination ports are encrypted and therefore useless to RSVP. This is not the case for authentication. For AH, the real header just follows the Authentication Header. So, it would be possible to use the real transport header for RSVP session definition and reservation.

To use the transport header, all that would need to be done is for the flow classifier to skip over AHs before classifying packets. No modification to RSVP formats or setup processing would be required. Applications would make reservations based on transport (i.e., UDP or

TCP) ports as usual.

The advantages of this approach are no changes to either IPSEC protocols or RSVP formats. The major disadvantage is that routers and hosts must skip all AHs before classifying packets. The working group decided that it was best to have a consistent solution for both AH and ESP.