Proposed RSVP Extensions for IPSEC IPv4 Data Flows

Lou Berger <lberger@fore.com>
Tim O'Malley <timo@bbn.com>

Topic of Discussion

- How to provide QoS to flows of secure IPv4 data with RSVP?

  Not how to provide security for RSVP messages!

- Internet Draft: draft-berger-rsvp-ext-02.txt
  - Authors: Berger, O'Malley, and Atkenson (Others contributed)
  - ftp://ds.internic.net/internet-drafts/draft-berger-rsvp-ext-02.txt

Problem Summary

RSVP Version 1 needs extension to support IPSEC data flows

- RSVP provides limited support for protocols that don't have UDP/TCP-link ports
- IPSEC headers do not contain TCP/UDP-like ports
  - Each header contains a four byte *SPI* field
    - SPI is a 32-bit pseudo-random value identifying the security association for a datagram
    - AH and ESP have SPI field at different header offsets
- Goal:
  - To provide QoS control to IPSEC data flows at a granularity similar to what can be provided to UDP and TCP flows.

Solution Summary

- Proposes:
  - Use SPI in place of UDP/TCP-like ports
    - SPIs used as *Generalized Source Ports* for FILTERing
  - Provide session demultiplexing via *Virtual Destination Ports*
    - *Virtual Destination Ports* are not carried in data packets
- Limitations:
  - Limits usage of WF style reservations
    - All flows with same protocol and destination address match WF reservations
  - Security impact on FF style reservations
    - Every time a new SPI is used, need to establish new reservation
    - May want to use SE to avoid problem
- Other options explored -- Use of SPIs was clear choice

Details of Extension

How it all works

- SPI Usage
- Session Definition
- Changes to Message Processing
- Object Definition

SPI Usage

- IPSEC assigns SPIs on a per destination address basis
  - Receiver selects SPI
  - SPI usually associated with a particular sender
    - Exception is for multicast traffic where SPI may be shared by multiple senders
- SPIs used exclusively for source port identification
  - Permits multiple senders to share single reservation
  - SPIs used for sender selection in FILTER_SPEC
  - Senders use SPIs in SENDER_TEMPLATEs

SPI Usage (continued)

- Data classified based on:
  (DestAddress, ProtocolId, SrcAddress, SPI)
  - Reservation style impact:
  - FF and SE - none
  - WF limitation
    - SrcAddress, and SPI are wildcards, so all flows with same protocol and destination address match WF reservations
    - Not likely to be issue since IPSEC users need to identify all valid senders and their associated SPIs
    - WF should be avoided for IPSEC data flows

SPI Usage (continued)

- Single new FILTER_SPEC and SENDER_TEMPLATE object type defined to support both AH and ESP
  - Message processing is unchanged
  - Only transport header offsets differ
- *Generalized Port Identifier* is used in object definition
  - Four byte field that contains SPI
  - Permits use of identical processing rules for non-IPSEC uses.

Session Definition

- Session = (DestAddress, ProtocolId, vDstPort)
  - DestAddress, ProtocolId - no change from current draft
  - vDstPort is a *Virtual Destination Port* that is used to distinguish between separate flows
    - Used for session demultiplexing
    - *Virtual Destination Ports* are not carried in data packets
- Why *vDstPort* ?
  - Need to separate flows for different reservations and end applications
  - SPI is only constant field in IPSEC transport headers
- SPI not suitable for use for destination based demultiplexing
  - SPI is unstructured so cannot be partitioned
  - SPIs typically assigned per sender, so would not be able to share reservations across senders

Changes to PATH Message Processing

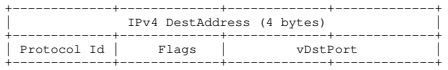
- Information should be recorded as currently specified
- New consistency checks:
  - SENDER_TEMPLATE type must match SESSION type
    - (not a new requirement)
  - ProtocolId in IPv4/GPI SESSION object must be of known type
    - RSVP/IPSEC implementations must support 51 (AH) or 50 (ESP)
  - vDstPort values of zero should result in an error

Changes to RESV Message Processing

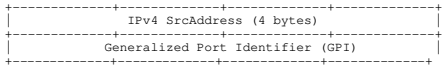
- New consistency checks:
  - FILTER_SPEC type must match type used in SESSION
    - (not a new requirement)
  - The GPI contained in the FILTER_SPEC must match the GPI contained in the SENDER_TEMPLATE
- Traffic control changes:
  - IPv4/GPI flows should be classified on:
    (DestAddress, ProtocolId, SrcAddress, GPI)
  - No UDP/TCP-like source or destination ports may be used
  - GPI transport header offset is +4 for AH, and +0 for ESP

Object Definition

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



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



- SENDER_TEMPLATE class = 11.
  - IPv4/GPI SENDER_TEMPLATE object: Class = 11, C-Type = 3
    Definition same as IPv4/GPI FILTER_SPEC object.

Open Issues

Four Issues:

- Use of *Generalized Port Identifiers* versus SPIs
- Handling of SPI changes
- Acceptance of proposal
- Publication form

Issue: *Generalized Port Identifier* versus SPI

- **Issue:** Should extensions allow for other protocols with same processing rules?
  - Other protocols may be supported using identical processing, but may require use of a different transport header offset
  - Processing of new objects are already protocol dependent
    - With AH, SPI is at transport header offset +4
    - With ESP, SPI is at transport header offset +0
- If no, should we define protocol specific objects?
  - Eliminates protocol specific processing
  - AH SESSION, FILTER_SPEC, and SENDER_TEMPLATE objects
    - Uses SPI at transport header offset +4
  - ESP SESSION, FILTER_SPEC, and SENDER_TEMPLATE objects
    - Uses SPI at transport header offset +0

Issue: Handling of SPI changes

- **Issue:** SPIs will change whenever the Security Association changes
  - Requires new SENDER_TEMPLATEs and FILTER_SPECs
  - Should not be the common case
- In current proposal:
  - SESSION definition is SPI independent - no issue
  - WF and SE filter styles multiple SPIs are supported - no issue
    - No impact on WF
    - With SE, new objects need to be sent but there will be no change in actual reserved resources
  - FF style possible - resources will be affected
    - Double resources allocated during transition period
    - Suggested alternative is to use multiple SE sessions, one per sender

Issue: Handling of SPI changes (continued)

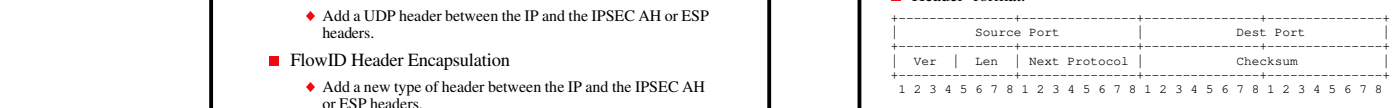
- Alternative proposal:
  - Extend SENDER_TEMPLATEs and FILTER_SPECs objects to include multiple SPIs (GPIs)
  - Impacts
    - Message processing
    - Data classifier

Considered Options

- UDP Encapsulation
  - Add a UDP header between the IP and the IPSEC AH or ESP headers.
- FlowID Header Encapsulation
  - Add a new type of header between the IP and the IPSEC AH or ESP headers.
- IPSEC modification
  - Modify IPSEC headers so that there are appropriate fields in same location as UDP and TCP ports.
- AH Transparency
  - Skip over the Authentication Header packet classifier processing.

FlowID Header Encapsulation

- This option was originally proposed by Greg Troxel
- Header format:



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