Note Well

This is a reminder of IETF policies in effect on various topics such as patents or code of conduct. It is only meant to point you in the right direction. Exceptions may apply. The IETF's patent policy and the definition of an IETF "contribution" and "participation" are set forth in BCP 79; please read it carefully.

As a reminder:

• By participating in the IETF, you agree to follow IETF processes and policies.
• If you are aware that any IETF contribution is covered by patents or patent applications that are owned or controlled by you or your sponsor, you must disclose that fact, or not participate in the discussion.
• As a participant in or attendee to any IETF activity you acknowledge that written, audio, video, and photographic records of meetings may be made public.
• Personal information that you provide to IETF will be handled in accordance with the IETF Privacy Statement.
• As a participant or attendee, you agree to work respectfully with other participants; please contact the ombudsteam ([https://www.ietf.org/contact/ombudsteam/](https://www.ietf.org/contact/ombudsteam/)) if you have questions or concerns about this.

Definitive information is in the documents listed below and other IETF BCPs. For advice, please talk to WG chairs or ADs:
• BCP 9 (Internet Standards Process)
• BCP 25 (Working Group processes)
• BCP 25 (Anti-Harassment Procedures)
• BCP 54 (Code of Conduct)
• BCP 78 (Copyright)
• BCP 79 (Patents, Participation)
• [https://www.ietf.org/privacy-policy/](https://www.ietf.org/privacy-policy/) (Privacy Policy)
Administrative Tasks

Bluesheets
We need volunteers to be:
• Two note takers
• One jabber scribe

Jabber: xmpp:ipsecme@jabber.ietf.org?join
MeetEcho: https://meetings.conf.meetecho.com/ietf109/?group=ipsecme&short=&item=1
Notes: https://codimd.ietf.org/notes-ietf-109-ipsecme
Agenda

• Note Well, technical difficulties and agenda bashing – Chairs (5 min) (16:00-16:05)
• Document Status – Chairs (5 min) (16:05-16:10)

• Work items
  • Labeled IPsec update – Paul Wouters (5 min) (16:10-16:15)
  • IP-TFS Update – Christian Hopps (10 min) (16:15-16:25)
  • YANG Model for IP Traffic Flow Security – Christian Hopps (5 min) (16:25-16:30)

• New items
  • Beyond 64KB limit of IKEv2 Payload – Valery Smyslov (10 min) (16:30-16:40)
  • IKEv2 Configuration for Encrypted DNS – Valery Smyslov (10 min) (16:40-16:50)
  • Revised Cookie Processing in IKEv2 – Valery Smyslov (15 min) (16:50-17:05)
  • Performance Enhancements for IPsec – Paul Wouters (20 min) (17:05-17:25)
  • IKEv1 graveyard – Paul Wouters (5 min) (17:25-17:30)
• AOB + Open Mic (17:30-18:00)
WG Status Report

In IETF Last Call

draft-ietf-ipsecme-ipv6-ipv4-codes

Work in progress:

draft-ietf-ipsecme-g-ikev2
draft-ietf-ipsecme-ikev2-intermediate
draft-ietf-ipsecme-ikev2-multiple-ke
draft-hopps-ipsecme-iptfs
draft-ietf-ipsecme-labeled-ipsec
Presentations

- **Labeled IPsec update** - Paul Wouters
- IP-TFS Update – Christian Hopps
- YANG Model for IP Traffic Flow Security – Christian Hopps
- Beyond 64KB limit of IKEv2 Payload – Valery Smyslov
- IKEv2 Configuration for Encrypted DNS – Valery Smyslov
- Revised Cookie Processing in IKEv2 – Valery Smyslov
- Performance Enhancements for IPsec – Paul Wouters
- IKEv1 graveyard – Paul Wouters
LABLELED IPSEC

IPsec, IETF 109
November 2019

Paul Wouters, RHEL Security
Minor fixups, added paragraph in Security Section and added Implementation Status

IETF 108 had Action Item for chairs: ”Perhaps go to WGLC”

Implemented in libreswan (will be in libreswan-4.2)
  • No interop testing done – anyone else implemented this?
  • Using private number 241

Was ready for WGLC. Now even more ready.

Request Early Code point?
Presentations

- Labeled IPsec update – Paul Wouters
- **IP-TFS Update** – Christian Hopps
- YANG Model for IP Traffic Flow Security – Christian Hopps
- Beyond 64KB limit of IKEv2 Payload – Valery Smyslov
- IKEv2 Configuration for Encrypted DNS – Valery Smyslov
- Revised Cookie Processing in IKEv2 – Valery Smyslov
- Performance Enhancements for IPsec – Paul Wouters
- IKEv1 graveyard – Paul Wouters
IP Traffic Flow Security
Improving IPsec Traffic Flow Confidentiality

IETF 109 – “draft-ietf-ipsecme-iptfs-03”
Update Since IETF 108

• Changes largely based on list discussions
• draft-ietf-ipsecme-iptfs-02 published Sept 30, 2020
  • Clarified fragment in following sequence number per WG feedback.
  • Add text highlighting ability to support zero-conf on receive.
    • Some WG discussion, should not be a MUST support (isn’t).
• draft-ietf-ipsecme-iptfs-03 published Nov 15, 2020 (IETF109)
  • Removed Zero-Conf functionality text
  • Removed IP protocol number assignment
  • Retain ESP Payload Type, assign value of 0x5
  • Congestion Control Updated
    • Change from “last received sequence number” to more standard “timestamp and echo”
    • Added “Transmission Delay” in addition to “Echo Delay”
Mailing List Discussions

- IP Number – Early Allocation Request
  - IETF 108 Benjamin (AD) indicated do request
  - Chair (Tero) still objects as too much trouble justifying
  - State based negotiation (e.g., IKE) is not the only use case of IPsec
  - Move to backup plan, just assign an ESP payload type of 0x5

- Zero-conf receive support
  - Simple to implement
  - Useful in non-IKE scenarios to simplify configuration (good Ops)
  - Controversial for some reason
  - Removed
Updated Congestion Control Payload Format

- Better matches RFC5348, identified as part of pre-TSV review and implementation
- TVal – Opaque timestamp from sender
- TEcho – Returned TVal to sender with Echo Delay indicating held time
- Echo Delay (21 bits) microseconds – Delta time from receiving TVal to sending in TEcho
- Transmit Delay (21 bits) microseconds – The current sending rate (packet delay)
  - Combined with local transmission delay to determine minimum RTT based on logical tunnel rate.
  - Required for fast packet paths where the in network RTT is smaller

```
| Sub-Type (1) | Reserved | BlockOffset |
| RTT          | Delay    |
| LossEventRate |
| LastSeqNum   |
| DataBlocks   |

```

```
| Sub-type (1) | Reserved | BlockOffset |
| RTT          | Echo Delay |
| LossEventRate |
| Transmit Delay |
| TVal         |
| TEcho        |
| DataBlocks   |
```

---

Updated Congestion Control Payload Format
Open Issues/Last Meeting Comments

• Transport Review (congestion control)
  • Suggested by Chair (Yoav) during IETF 108
  • Latest update based on implementation experience
    • Previous version worked fine, but was overly clever and restricting
  • Had meeting with David Black, ready to move on this
Other Notes

• Open source implementation
  • Implemented in VPP and Strongswan
  • Congestion Control Supported
  • IKEv2 Supported
  • In publication process now – hoping to release next month

• Open to collaboration/interoperability testing.
Moving Forward

- All issues raised by WG addressed in current version
- Transport review seems the remaining action
- As part of WGLC?
Questions and Comments
Backup Slides
Comparison Data
Why is this Needed?

- Current Solution: ESP + Padding 1:1
- Not Deployable.

Solution Cost (I-Mix)

<table>
<thead>
<tr>
<th></th>
<th>ESP + Pad</th>
<th>IPTFS</th>
<th>Enet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bandwidth Used</td>
<td>1Gb</td>
<td>1Gb</td>
<td>1Gb</td>
</tr>
</tbody>
</table>

| I-Mix Throughput | 219Mb | 943Mb | 672Mb |

Bandwidth Efficiency (I-Mix)
# Overhead Comparison in Octets

<table>
<thead>
<tr>
<th>L3 MTU</th>
<th>PSize</th>
<th>ESP+Pad</th>
<th>ESP+Pad</th>
<th>ESP+Pad</th>
<th>IP-TFS</th>
<th>IP-TFS</th>
<th>IP-TFS</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>500</td>
<td>1424</td>
<td>8924</td>
<td>3.0</td>
<td>1.1</td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>128</td>
<td>412</td>
<td>1336</td>
<td>8836</td>
<td>9.6</td>
<td>3.5</td>
<td>0.6</td>
<td></td>
</tr>
<tr>
<td>256</td>
<td>284</td>
<td>1208</td>
<td>8708</td>
<td>19.1</td>
<td>7.0</td>
<td>1.1</td>
<td></td>
</tr>
<tr>
<td>536</td>
<td>4</td>
<td>928</td>
<td>8428</td>
<td>40.0</td>
<td>14.7</td>
<td>2.4</td>
<td></td>
</tr>
<tr>
<td>576</td>
<td>576</td>
<td>888</td>
<td>8388</td>
<td>43.0</td>
<td>15.8</td>
<td>2.6</td>
<td></td>
</tr>
<tr>
<td>1460</td>
<td>268</td>
<td>4</td>
<td>7504</td>
<td>109.0</td>
<td>40.0</td>
<td>6.5</td>
<td></td>
</tr>
<tr>
<td>1500</td>
<td>228</td>
<td>1500</td>
<td>7464</td>
<td>111.9</td>
<td>41.1</td>
<td>6.7</td>
<td></td>
</tr>
<tr>
<td>8960</td>
<td>1408</td>
<td>1540</td>
<td>4</td>
<td>668.7</td>
<td>245.5</td>
<td>40.0</td>
<td></td>
</tr>
<tr>
<td>9000</td>
<td>1368</td>
<td>1500</td>
<td>9000</td>
<td>671.6</td>
<td>246.6</td>
<td>40.2</td>
<td></td>
</tr>
</tbody>
</table>
# Overhead as Percentage of Inner Packet

<table>
<thead>
<tr>
<th>Type</th>
<th>ESP+Pad</th>
<th>ESP+Pad</th>
<th>ESP+Pad</th>
<th>IP-TFS</th>
<th>IP-TFS</th>
<th>IP-TFS</th>
</tr>
</thead>
<tbody>
<tr>
<td>MTU</td>
<td>576</td>
<td>1500</td>
<td>9000</td>
<td>576</td>
<td>1500</td>
<td>9000</td>
</tr>
<tr>
<td>PSize</td>
<td>540</td>
<td>1464</td>
<td>8964</td>
<td>536</td>
<td>1460</td>
<td>8960</td>
</tr>
<tr>
<td>40</td>
<td>1250.0%</td>
<td>3560.0%</td>
<td>22310.0%</td>
<td>7.46%</td>
<td>2.74%</td>
<td>0.45%</td>
</tr>
<tr>
<td>128</td>
<td>321.9%</td>
<td>1043.8%</td>
<td>6903.1%</td>
<td>7.46%</td>
<td>2.74%</td>
<td>0.45%</td>
</tr>
<tr>
<td>256</td>
<td>110.9%</td>
<td>471.9%</td>
<td>3401.6%</td>
<td>7.46%</td>
<td>2.74%</td>
<td>0.45%</td>
</tr>
<tr>
<td>536</td>
<td>0.7%</td>
<td>173.1%</td>
<td>1572.4%</td>
<td>7.46%</td>
<td>2.74%</td>
<td>0.45%</td>
</tr>
<tr>
<td>576</td>
<td>100.0%</td>
<td>154.2%</td>
<td>1456.2%</td>
<td>7.46%</td>
<td>2.74%</td>
<td>0.45%</td>
</tr>
<tr>
<td>1460</td>
<td>18.4%</td>
<td>0.3%</td>
<td>514.0%</td>
<td>7.46%</td>
<td>2.74%</td>
<td>0.45%</td>
</tr>
<tr>
<td>1500</td>
<td>15.2%</td>
<td>100.0%</td>
<td>497.6%</td>
<td>7.46%</td>
<td>2.74%</td>
<td>0.45%</td>
</tr>
<tr>
<td>8960</td>
<td>15.7%</td>
<td>17.2%</td>
<td>0.0%</td>
<td>7.46%</td>
<td>2.74%</td>
<td>0.45%</td>
</tr>
<tr>
<td>9000</td>
<td>15.2%</td>
<td>16.7%</td>
<td>100.0%</td>
<td>7.46%</td>
<td>2.74%</td>
<td>0.45%</td>
</tr>
</tbody>
</table>
## Bandwidth Utilization over Ethernet

<table>
<thead>
<tr>
<th>Size</th>
<th>Enet</th>
<th>ESP</th>
<th>E + P</th>
<th>E + P</th>
<th>E + P</th>
<th>IPTFS</th>
<th>IPTFS</th>
<th>IPTFS</th>
</tr>
</thead>
<tbody>
<tr>
<td>38</td>
<td>any</td>
<td>any</td>
<td>590</td>
<td>1514</td>
<td>9014</td>
<td>590</td>
<td>1514</td>
<td>9014</td>
</tr>
<tr>
<td>40</td>
<td>47.6%</td>
<td>35.1%</td>
<td>6.5%</td>
<td>2.6%</td>
<td>0.4%</td>
<td>87.3%</td>
<td>94.9%</td>
<td>99.1%</td>
</tr>
<tr>
<td>128</td>
<td>77.1%</td>
<td>63.4%</td>
<td>20.8%</td>
<td>8.3%</td>
<td>1.4%</td>
<td>87.3%</td>
<td>94.9%</td>
<td>99.1%</td>
</tr>
<tr>
<td>256</td>
<td>87.1%</td>
<td>77.6%</td>
<td>41.7%</td>
<td>16.6%</td>
<td>2.8%</td>
<td>87.3%</td>
<td>94.9%</td>
<td>99.1%</td>
</tr>
<tr>
<td>536</td>
<td>93.4%</td>
<td>87.9%</td>
<td>87.3%</td>
<td>34.9%</td>
<td>5.9%</td>
<td>87.3%</td>
<td>94.9%</td>
<td>99.1%</td>
</tr>
<tr>
<td>576</td>
<td>93.8%</td>
<td>88.6%</td>
<td>46.9%</td>
<td>37.5%</td>
<td>6.4%</td>
<td>87.3%</td>
<td>94.9%</td>
<td>99.1%</td>
</tr>
<tr>
<td>1460</td>
<td>97.5%</td>
<td>95.2%</td>
<td>79.3%</td>
<td>94.9%</td>
<td>16.2%</td>
<td>87.3%</td>
<td>94.9%</td>
<td>99.1%</td>
</tr>
<tr>
<td>1500</td>
<td>97.5%</td>
<td>95.3%</td>
<td>81.4%</td>
<td>48.8%</td>
<td>16.6%</td>
<td>87.3%</td>
<td>94.9%</td>
<td>99.1%</td>
</tr>
<tr>
<td>8960</td>
<td>99.6%</td>
<td>99.2%</td>
<td>81.1%</td>
<td>83.2%</td>
<td>99.1%</td>
<td>87.3%</td>
<td>94.9%</td>
<td>99.1%</td>
</tr>
<tr>
<td>9000</td>
<td>99.6%</td>
<td>99.2%</td>
<td>81.4%</td>
<td>83.6%</td>
<td>49.8%</td>
<td>87.3%</td>
<td>94.9%</td>
<td>99.1%</td>
</tr>
</tbody>
</table>
Latency

- Latency values seem very similar
- IP-TFS values represent max latency
- IP-TFS provides for constant high bandwidth
- ESP + padding value represents min latency
- ESP + padding often greatly reduces available bandwidth.

<table>
<thead>
<tr>
<th></th>
<th>ESP+Pad 1500</th>
<th>ESP+Pad 9000</th>
<th>IP-TFS 1500</th>
<th>IP-TFS 9000</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>1.14 us</td>
<td>7.14 us</td>
<td>1.17 us</td>
<td>7.17 us</td>
</tr>
<tr>
<td>128</td>
<td>1.07 us</td>
<td>7.07 us</td>
<td>1.10 us</td>
<td>7.10 us</td>
</tr>
<tr>
<td>256</td>
<td>0.97 us</td>
<td>6.97 us</td>
<td>1.00 us</td>
<td>7.00 us</td>
</tr>
<tr>
<td>536</td>
<td>0.74 us</td>
<td>6.74 us</td>
<td>0.77 us</td>
<td>6.77 us</td>
</tr>
<tr>
<td>576</td>
<td>0.71 us</td>
<td>6.71 us</td>
<td>0.74 us</td>
<td>6.74 us</td>
</tr>
<tr>
<td>1460</td>
<td>0.00 us</td>
<td>6.00 us</td>
<td>0.04 us</td>
<td>6.04 us</td>
</tr>
<tr>
<td>1500</td>
<td>1.20 us</td>
<td>5.97 us</td>
<td>0.00 us</td>
<td>6.00 us</td>
</tr>
</tbody>
</table>
Transport Mode

- Motivation is common GRE/IPsec-Transport Use
- Some interest in generic transport mode.
- What IP header fields to support
  - Simple
    - No fields – GRE Support
      - If the packet header is different then the last, pad current IPTFS out and start new one
      - If is inefficient due to frequent header differences, then use tunnel mode.
  - All Fields
    - IP header replicated inside payload for each packet
      - Similar to tunnel mode, but less efficient.
  - Complex
    - IP Header compression Ideas (deviations, etc)
      - Complex solution in need of a problem?
- Enough separable work to publish as a separate document.
Presentations

- Labeled IPsec update – Paul Wouters
- IP-TFS Update – Christian Hopps
- YANG Model for IP Traffic Flow Security – Christian Hopps
- Beyond 64KB limit of IKEv2 Payload – Valery Smyslov
- IKEv2 Configuration for Encrypted DNS – Valery Smyslov
- Revised Cookie Processing in IKEv2 – Valery Smyslov
- Performance Enhancements for IPsec – Paul Wouters
- IKEv1 graveyard – Paul Wouters
YANG Model for IP Traffic Flow Security

IETF 109 – “draft-fedyk-ipsecme-yang-iptfs-01”
Changes since IETF108

• Some reminders
  • Draft objective -- YANG support for IP-TFS
    • Expect to also do a derivative SNMP draft
  • Draft approach – Augment existing IPsec YANG model
    • ietf-i2nsf-sdn-ipsec-flow-protection

• Open issue with base YANG model discussed at last meeting resolved
  • Base yang model focused on controller use case
  • Previous version was incompatible with device configuration
  • Based on comments, incompatible usage was made a YANG feature
    • Now usable as foundation for TFS device configuration
  • Draft updated to align with ietf-i2nsf-sdn-ipsec-flow-protection changes
More details on draft-ietf-i2nsf-sdn-ipsec-flow-protection

- I2NSF WG defined SDN model provides for an IKE and IKE-less operation
- IKE module intentionally missing a Security Association Database
  - Reason given: centralized controller (SDN) doesn't care about SAs
  - Has child-sa-info to hold connections SA related info
- IKE module missing SA information
  - child-sa-info only has pfs-groups and lifetime values
  - no information on selected transforms, etc
- Existing model (IKE/IKE-less) does not have Basic IPsec counters
- IP-TFS YANG augments this model
IP-TFS Configuration

- Congestion Control
  - Boolean

- Packet Size (L3 Packet size)
  - Fixed Size
  - Use Path MTU (set or lowers fixed)

- Bit rate
  - L3 Bit rate or
  - L2 Bit rate

- Allow fragmentation
  - Of Inner packets using data blocks and IP TFS offsets

Packet Transmission Frequency = Bit rate/Packet size

Note these are minimal controls vendors or future work may augment
Operational Statistics

- **Outer IPsec Packet – IPsec Counters**
  - tx IPsec packets and octets
  - rx IPsec packets and octets
  - rx dropped packet counts
  - rx error counts/type

- **Inner IP Packets – IP-TFS Counters**
  - tx packets and octets
  - tx extra pad packets and octets
  - tx all pad packets and octets
  - rx packets and octets
  - rx extra pad packets and octets
  - rx all pad packets and octets
  - rx errored packets
  - rx missed packets
  - rx incomplete inner packets

\[
\text{IP-TFS Protocol Overhead} = \text{Outer Packet Octets} - \text{Inner Packet Octets} - \text{Pad Octets}
\]
Next Steps

• Authors request WG adoption
Comments / Questions?
More Details
module: ietf-ipsecme-iptfs
  augment /nsfike:ipsec-ike/nsfike:conn-entry
    /nsfike:spd/nsfike:spd-entry
    /nsfike:ipsec-policy-config
    /nsfike:processing-info/nsfike:ipsec-sa-cfg:
      +-rw traffic-flow-security
      |   +-rw congestion-control?   boolean
      |   +-rw packet-size
      |     |   +-rw use-path-mtu?       boolean
      |     |   +-rw outer-packet-size?  uint16
      |     |   +-rw (tunnel-rate)?
      |     |     |   +(12-bitrate)
      |     |     |   +(13-bitrate)
      |     |     |   +-rw 12-bitrate?          uint64
      |     |     |   +-rw 13-bitrate?          uint64
      |     +-rw dont-fragment?      boolean

augment /nsfike:ipsec-ike/nsfike:conn-entry
  /nsfike:child-sa-info:
    +-ro traffic-flow-security
    |   +-ro congestion-control?   boolean
    |   +-ro packet-size
    |     |   +-ro use-path-mtu?       boolean
    |     |   +-ro outer-packet-size?  uint16
    |     |   +-ro (tunnel-rate)?
    |     |     |   +(12-bitrate)
    |     |     |   +(13-bitrate)
    |     |     |   +-ro 12-bitrate?         uint64
    |     |     |   +-ro 13-bitrate?         uint64
    |     +-ro dont-fragment?      boolean
IP-TFS Config augment nfs-ikels

User Provided Config
(same as IKE, under spd-entry grouping)

Operational (Actual) Config
(diff from IKE, now under SAD entry)
Statistics augment ipsec-ike (all-new)

```
augment /nsfike:ipsec-ike/nsfike:conn-entry/nsfike:child-sa-info:
  +--ro ipsec-stats {ipsec-stats}?
     |    +--ro tx-packets? uint64
     |    +--ro tx-octets? uint64
     |    +--ro tx-drop-packets? uint64
     |    +--ro rx-packets? uint64
     |    +--ro rx-octets? uint64
     |    +--ro rx-drop-packets? uint64
  +--ro iptfs-stats {iptfs-stats}?
     |    +--ro tx-inner-packets? uint64
     |    +--ro tx-inner-octets? uint64
     |    +--ro tx-extra-pad-packets? uint64
     |    +--ro tx-extra-pad-octets? uint64
     |    +--ro tx-all-pad-packets? uint64
     |    +--ro tx-all-pad-octets? uint64
     |    +--ro rx-inner-packets? uint64
     |    +--ro rx-inner-octets? uint64
     |    +--ro rx-extra-pad-packets? uint64
     |    +--ro rx-extra-pad-octets? uint64
     |    +--ro rx-all-pad-packets? uint64
     |    +--ro rx-all-pad-octets? uint64
     |    +--ro rx-errored-packets? uint64
     |    +--ro rx-missed-packets? uint64
     |    +--ro rx-incomplete-inner-packets? uint64
```

IPsec Statistics

IP-TFS Statistics
Statistics augment ipsec-ikeless (all-new)

```
augment /nsfikels:ipsec-ikeless/nsfikels:sad/nsfikels:sad-entry:
  +--rw ipsec-stats {ipsec-stats}?
    |    +--ro tx-packets?       uint64
    |    +--ro tx-octets?        uint64
    |    +--ro tx-drop-packets?  uint64
    |    +--ro rx-packets?       uint64
    |    +--ro rx-octets?        uint64
    |    +--ro rx-drop-packets?  uint64
  +--rw iptfs-stats {iptfs-stats}?
    |    +--ro tx-inner-packets? uint64
    |    +--ro tx-inner-octets?  uint64
    |    +--ro tx-extra-pad-packets? uint64
    |    +--ro tx-extra-pad-octets? uint64
    |    +--ro tx-all-pad-packets? uint64
    |    +--ro tx-all-pad-octets? uint64
    |    +--ro rx-inner-packets?  uint64
    |    +--ro rx-inner-octets?  uint64
    |    +--ro rx-extra-pad-packets? uint64
    |    +--ro rx-extra-pad-octets? uint64
    |    +--ro rx-all-pad-packets? uint64
    |    +--ro rx-all-pad-octets? uint64
    |    +--ro rx-errorred-packets? uint64
    |    +--ro rx-missed-packets?  uint64
    |    +--ro rx-incomplete-inner-packets? uint64
```

IPsec Statistics

IP-TFS Statistics
IP –TFS Tunnel Mode Packets - Summary

L3 Packet size = IPsec Outer Packet size

L2 Packet size

Inner packet size

Inner packet size

Padding size
Presentations

- Labeled IPsec update – Paul Wouters
- IP-TFS Update – Christian Hopps
- YANG Model for IP Traffic Flow Security – Christian Hopps
- **Beyond 64KB limit of IKEv2 Payload** – Valery Smyslov
- IKEv2 Configuration for Encrypted DNS – Valery Smyslov
- Revised Cookie Processing in IKEv2 – Valery Smyslov
- Performance Enhancements for IPsec – Paul Wouters
- IKEv1 graveyard – Paul Wouters
Large Payloads in IKEv2

draft-tjhai-ikev2-beyond-64k-limit

CJ Tjhai (Post-Quantum)
Tobias Heider (genua GmbH)
Valery Smyslov (ELVIS-PLUS)

IETF 109
Motivation

- draft-ietf-ipsecme-ikev2-multiple-ke addresses issues of using large keys for Key Exchange methods (common in PQC) in IKEv2
- This draft still limits the size of any single public key to 64K – the maximum size of IKEv2 payload
  - most NIST Third Round Candidate Algorithms fit into this restriction
- However, some national regulators (e.g. BSI) recommends using Classic McElice PQKE which smallest public-key is 255KB (while more conservative parameter sets are even around 1 MB)
- Using post-quantum signatures and post-quantum certificates (draft-ounsworth-pq-pq-composite-sigs) may lead to the situation when AUTH and CERT payloads also grow beyond 64K
Goals

• The goal of the document is to define a way for using some specific data blobs in IKEv2 if they grow beyond 64K
  – public keys for key exchange methods (KE)
  – signatures (AUTH)
  – certificates (CERT)
• The defined mechanism must be backward compatible
• Reliability of transferring large data in IKEv2 should be addressed
• The defined mechanism must be simple and must introduce minimal changes to IKEv2
Not Goal

- There is no goal to define a generic mechanism for IKEv2 which would allow any payload be greater than 64K
Proposed Approach

- If amount of data doesn’t fit into a single payload then split data into chunks less than 64K and put them into a sequence of payloads with the same type; receiving side will concatenate data from a sequence of payloads having the same type
  - this approach works well if only one payload of this type can appear in the message according to IKEv2 (true for KE and AUTH, not true for CERT, but can be worked around)
  - if such sequence of payloads appears inside Encrypted payload (true for AUTH, CERT), then the Length field of the Encrypted payload cannot be used, but this doesn’t matter, since the length of Encrypted payload can always be deduced from the length of IKE message
Example

**Initiator**

**IKE_SA_INIT**
HDR, SAI1, KEi1, Ni

**IKE_INTERMEDIATE**
HDR, SK{KE2i, KE2i, KE2i}

**IKE_AUTH**
HDR, SK{IDi, [CERT, CERT, CERT,] [CERTREQ,] [IDr,] AUTH, AUTH, SAI2, TSi, TSr}

---

**Responder**

**IKE_SA_INIT**
HDR, SAR1, KE1r, Nr, [CERTREQ,]

**IKE_INTERMEDIATE**
HDR, SK{KE2r, KE2r}

**IKE_AUTH**
HDR, SK{IDr, [CERT, CERT,] AUTH, AUTH, SAI2, TSi, TSr}
Discussion

- The proposed approach is simple and easy to implement
- It doesn’t touch IKE state machine and doesn’t change sequence of exchanges
- It allows amount of data to transfer to be very different in different directions (very important for some KEMs)
- The proposed approach does require some tweaks (like handling some payloads differently than others)
  - that is that…
- IKE messages will grow in size making it difficult to use UDP to transport them
  - it is anticipated that TCP (or some other reliable transport) will often be used in this case
Thanks

• Comments? Questions?
• Is this problem worth to address?
• Is the suggested approach reasonable?
• WG adoption?
Presentations

- Labeled IPsec update – Paul Wouters
- IP-TFS Update – Christian Hopps
- YANG Model for IP Traffic Flow Security – Christian Hopps
- Beyond 64KB limit of IKEv2 Payload – Valery Smyslov
- IKEv2 Configuration for Encrypted DNS – Valery Smyslov
- Revised Cookie Processing in IKEv2 – Valery Smyslov
- Performance Enhancements for IPsec – Paul Wouters
- IKEv1 graveyard – Paul Wouters
IKEv2 Configuration for Encrypted DNS

draft-btw-add-ipsecmme-ike-01

Mohamed Boucadair (Orange)
Tirumaleswar Reddy (McAfee, Inc.)
Dan Wing (Citrix Systems, Inc.)
Valery Smyslov (ELVIS-PLUS)

November 2020, IETF#109
Status

• Presented at IETF#108
• Comments raised so far:
  – Complexity induced by muxing the attributes (mask bit)
  – Check if there are DoQ specifics
  – Supply DoH URI Template
• Fixed in 01: See next slide
• There are trade-offs
Changes from -00

- New Attribute format for Encrypted DNS
  - separate attribute types for each Encrypted DNS type (DoT, DoH, DoQ) and for IP version
    - ENCDNS_IP4_DOT, ENCDNS_IP6_DOT
    - ENCDNS_IP4_DOH, ENCDNS_IP6_DOH
    - ENCDNS_IP4_DOQ, ENCDNS_IP6_DOQ
  - port number is added
    - Triggered by a check with the authors of DoQ to assess if they have specific configuration data to be returned to DoQ clients
  - “scope” bit is removed
Separate Attribute Types

• We assume all types of Encrypted DNS are equivalent, so the client can be configured with any of them
  – with this approach the client includes all attributes for Encrypted DNS types it supports and the server returns back one (or few) of them with the DNS server(s) details
  – each attribute can contain several IP addresses of resolvers
Port Number

- Support for customizing port number for DNS servers is added
  - Port number is the same for all IP-addresses in a single attribute
  - If DNS servers have different port numbers, then separate attributes of the same type should be returned
Scope

• “Scope” bit is removed
  – DNS server selected by the client outside of the VPN tunnel is out of scope of this draft
## Attribute Format

<table>
<thead>
<tr>
<th>R</th>
<th>Attribute Type</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Port Number</td>
<td>RESERVED</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>~</td>
<td>IP Addresses</td>
<td>~</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>~</td>
<td>DNS Authentication Domain Name</td>
<td>~</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
DoH Specifics

• DoH servers may support more than one URI Template
• The DoH server may also host several DoH services (e.g., no-filtering, blocking adult content)
  – These services can be discovered as templates
• The client uses a well-known URI "resinfo" to discover these templates:
  
  ```
  https://doh.example.com/.well-known/resinfo
  ```

  Authentication Domain Name To be assigned by IANA

• Discovering the well-known URI is out of scope of this draft and is discussed in Section 5 of draft-btw-add-home
• Draft will use whatever mechanism(s) are finalized by the ADD WG for URI template discovery
Next Steps

- Comments?
- Questions?
- Consider WG adoption

Thank you
Presentations

• Labeled IPsec update – Paul Wouters
• IP-TFS Update – Christian Hopps
• YANG Model for IP Traffic Flow Security – Christian Hopps
• Beyond 64KB limit of IKEv2 Payload – Valery Smyslov
• IKEv2 Configuration for Encrypted DNS – Valery Smyslov
• Revised Cookie Processing in IKEv2 – Valery Smyslov
• Performance Enhancements for IPsec – Paul Wouters
• IKEv1 graveyard – Paul Wouters
Revised Cookie Processing in IKEv2

draft-smyslov-ipsecme-ikev2-cookie-revised

Valery Smyslov
svan@elvis.ru

IETF 109
Using Cookies in IKEv2

<table>
<thead>
<tr>
<th>Initiator</th>
<th>Responder</th>
</tr>
</thead>
<tbody>
<tr>
<td>req1 <strong>IKE_SA_INIT</strong> HDR,SA11,KEi,Ni</td>
<td>resp1 <strong>IKE_SA_INIT</strong> HDR,N(COOKIE)</td>
</tr>
</tbody>
</table>

req2 **IKE_SA_INIT** HDR,N(COOKIE),SA11,KEi,Ni

req3 **IKE_AUTH** HDR,SK{IDi,[CERT,][CERTREQ,] [IDr,] AUTH, SAi2, TSi, TSr} 

The most recent IKE_SA_INIT request is included in the AUTH payload calculation in the IKE_AUTH exchange. In this example it is req2 for both the initiator and the responder.
Problem Scenario 1

The most recent IKE_SA_INIT request sent by the initiator is req2, while the responder only received req1, so authentication is failed.
Problem Scenario 2

<table>
<thead>
<tr>
<th>Initiator</th>
<th>Responder</th>
</tr>
</thead>
<tbody>
<tr>
<td>req1 <strong>IKE_SA_INIT</strong> HDR,SAi1,KEi,Ni</td>
<td>Under attack resp1 <strong>IKE_SA_INIT</strong> HDR,N(COOKIE, c1)</td>
</tr>
<tr>
<td>req1 (resend) <strong>IKE_SA_INIT</strong> HDR,SAi1,KEi,Ni</td>
<td>Under attack, cookie secret changed resp2 <strong>IKE_SA_INIT</strong> HDR,N(COOKIE, c2)</td>
</tr>
<tr>
<td>req2 <strong>IKE_SA_INIT</strong> HDR,N(COOKIE, c2),SAi1,KEi,Ni</td>
<td>resp3 <strong>IKE_SA_INIT</strong> HDR,SAr1,KEr,Nr,[CERTREQ,] X</td>
</tr>
<tr>
<td>req3 <strong>IKE_SA_INIT</strong> HDR,N(COOKIE, c1),SAi1,KEi,Ni</td>
<td></td>
</tr>
<tr>
<td>req4 <strong>IKE_AUTH</strong> HDR,SK{[IDi,[CERT,][CERTREQ,][IDr,] AUTH, SAi2, TSi, TSr}</td>
<td>resp4 <strong>IKE_AUTH</strong> HDR,SK{N(AUTHENTICATION_FAILED)}</td>
</tr>
</tbody>
</table>

The most recent **IKE_SA_INIT** request sent by the initiator is req3, while the responder only received req2, so authentication is failed.
Source of the Problem

- The IKE_SA_INIT request can be sent several times with different content depending on the responder state.
- If there is a high probability of packets loss and reordering, then peers may complete the IKE_SA_INIT exchange having different views on what was the most recently sent IKE_SA_INIT request.
- This request message is used in calculation of the AUTH payload, so if peers use different messages authentication would erroneously fail.
Severity of the Problem

• There are some preconditions for this problem to become noticeable
  – network with high probability of packet loss and delay
  – relatively frequent change of responder state (either changing cookie generation secret or changing responder’s mind whether it is under attack)
• It might be rare in normal conditions, but in stress tests we observed that up to 5% of SAs failed due to this problem
  – for customers it looks strange that authentication sometimes failed with proper credentials
• This is a protocol flaw
Proposed Solution Overview

- Revise cookie processing by excluding Notify payload containing cookie (if present) from the IKE_SA_INIT request message when calculating the AUTH payload content
  - the cookie is already verified by the responder, no need to include it into the data to be authenticated
- For backward compatibility make the revised processing negotiable
Responder includes a new notification REVISED_COOKIE in the message containing COOKIE notification. If initiator also supports this extension, it returns cookie in this notification instead of COOKIE notification.
Revised Cookie Processing

• If peers agreed upon using this extension then the cookie processing is changed
  – no changes in cookie anti-clogging function – responder still sends stateless cookie and when it is returned back by initiator it MUST be verified before message is processed

According to RFC7296 initiator’s AUTH payload is calculated by signing (or MAC’ing) the blob:

\[ \text{InitiatorSignedOctets} = \text{RealMessage1 | NonceRData | MACedIDForI} \]

– if REVISED_COOKIE Notify payload is present in RealMessage1 (i.e. in IKE_SA_INIT request message), then for the purpose of AUTH payload calculation the message is modified as if it contained no this payload
### Adjusting IKE_SA_INIT Request for AUTH Payload Calculation

<table>
<thead>
<tr>
<th>IKE SA Initiator's SPI</th>
<th>IKE SA Responder's SPI</th>
</tr>
</thead>
<tbody>
<tr>
<td>NextPld1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Message ID</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MsgLen</td>
</tr>
<tr>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td>PldLen1</td>
</tr>
<tr>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Cookie</td>
</tr>
<tr>
<td></td>
<td>Rest of Message</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IKE SA Initiator's SPI</th>
<th>IKE SA Responder's SPI</th>
</tr>
</thead>
<tbody>
<tr>
<td>NextPld2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Message ID</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MsgLen’ = MsgLen - PldLen1</td>
</tr>
<tr>
<td></td>
<td>Rest of Message</td>
</tr>
</tbody>
</table>

**REVISED_COOKIE Notify Payload**
Thanks

• Comments? Questions?
• Is this problem worth to address?
• Is the suggested approach reasonable?
• WG adoption?
Presentations

- Labeled IPsec update – Paul Wouters
- IP-TFS Update – Christian Hopps
- YANG Model for IP Traffic Flow Security – Christian Hopps
- Beyond 64KB limit of IKEv2 Payload – Valery Smyslov
- IKEv2 Configuration for Encrypted DNS – Valery Smyslov
- Revised Cookie Processing in IKEv2 – Valery Smyslov
- **Performance Enhancements for IPsec – Paul Wouters**
- IKEv1 graveyard – Paul Wouters
IKEV2 SUPPORT FOR PER-QUEUE CHILD SA

IPsec, IETF 109
November 2020

Antony Antony, Steffen Klassert, Paul Wouters
Current IPsec SA limitation

- An IPsec SA implementation typically can only use 1 CPU
- An IPsec SA implementation typically can have only 1 QoS
- Launching multiple IPsec SAs is possible, but can lead to interoperability issues:
  - Duplicate IPsec SAs getting deleted as “old”
  - Disagreement about how many IPsec SAs to use leading to TS_UNACCEPTABLE errors until optimum found
- QoS requires both sides to signal QoS level per IPsec SA
Resolve limitation by:

- Give implementation advise on how to handle multiple IPsec SA’s with identical Traffic Selectors (please review document)
- Two new NOTIFY payloads for IPsec SA
  - NUM_QUEUES(pref, max)
  - QUEUE_INFO(opaque)
Implementation Status:

- Linux kernel XFRM implementation (Steffen Klassert)
  - Including per-cpu (on-demand) ACQUIRE messages
- Libreswan implementation (Antony Antony)
  - Basic: implements preconfigured number of IPsec SAs
- Strongswan implementation (Antony Antony)
  - Basic: implements preconfigured number of IPsec SAs
- See draft Implementation Status for links to software
Benchmarks

- pCPU max 22.3 Gbps
- One SA max 3.9 Gbps
- clear text 34.4 Gbps

Throughput Gbits/sec vs #Flows
Open Issues for IKE

- Is NUM(preferred, max) the right negotiation?
- Is there value (and/or danger) in signaling CPUID?
- Would QUEUE_INFO need a sub registry?
- Corner cases (eg both ends initiate for final slot)?
- IPsec rekey changes SPI, might change CPU affinity
- NAT mapping updates causing RSS hashing changes
Hardware (issues)

• Sender assumed to use different CPUs (e.g., server with threads)
• Receiver hardware is where real support is needed
• Network card support for RSS
  • RSS usually only supports UDP/TCP port hashing selector
  • RSS support for ESP if there, often incomplete/lacking
  • n-tuple support – rarely available for SPI selector
  • n-tuple – if available, requires ‘manual’ configuration
  • Virtual NIC support ongoing (RSS, RFS/aRFS, “multinic”)
• Better and standardized hardware support would be good
Feedback

• Any questions?
• Is there interest in the WG?
• Especially interested to hear from HW vendors
To use the references linux / libreswan / strongswan, you need to have support for one of these:

- NIC with RSS for ESP support
- NIC with RSS support with enabling UDP encap (usually done by lying in NAT_DETECTION_* payloads)
- NIC with n-tuple support for ESP, eg:

```
ethtool --config-ntuple eth0 flow-type esp4 src-ip \ 
192.168.1.1 dst-ip 192.168.1.101 spi 0x12345678 \ 
action 1 loc 2
(ideally with n-tuple SPI selector for ESPinUDP)
```

- NIC with n-tuple support for UDP, using UDP encap ESP

```
ethtool --config-ntuple eth0 flow-type ip4 src-ip \ 
192.168.1.1 dst-ip 192.168.1.101 action 1 loc 2
```
Presentations

- Labeled IPsec update – Paul Wouters
- IP-TFS Update – Christian Hopps
- YANG Model for IP Traffic Flow Security – Christian Hopps
- Beyond 64KB limit of IKEv2 Payload – Valery Smyslov
- IKEv2 Configuration for Encrypted DNS – Valery Smyslov
- Revised Cookie Processing in IKEv2 – Valery Smyslov
- Performance Enhancements for IPsec – Paul Wouters
- IKEv1 graveyard – Paul Wouters
IKEV1 GRAVEYARD

IPsec, IETF 109
November 2019

Paul Wouters, RHEL Security
1. Tells people to stop using IKEv1 – moves RFC 2409 to Historic
2. Mark IKEv1 era MAY algorithms as deprecated in IANA for IKEv2/ESP
3. Need an RFC for the “deprecated” column at IANA
4. Does not provide updated algorithm guidelines as in the RFC 8221 / 8247 series
5. IETF108 had Action Item for chairs: ”Perhaps go to WGLC” (but document has not yet been adopted?)
Open Discussion

• Other points of interest?