Tunnel Fragmentation

IETF94 Intarea Working Group
November 5, 2015
Fred L. Templin (fred.l.templin@boeing.com)
https://datatracker.ietf.org/doc/draft-templin-intarea-grefrag/
https://datatracker.ietf.org/doc/draft-herbert-gue-fragmentation/
Fragmenting Tunneled Packets

• Tunneled packet consists of an encapsulation IP header, followed by an encapsulation shim, followed by an encapsulated IP packet:

• Fragmentation sometimes unavoidable
Fragmentation Alternative 1: Fragment Encapsulation Packet

- Break encapsulation packet into N pieces
- Append an identical IP header at the beginning of each piece
- All fragments except the final fragment have MF=1
Alternative 1 Issues

- IPv4 Identification field is only 16 bits – means that ID value could wrap around and cause fragment misassociations even at moderate data rates (RFC4963; RFC6864) - NAT makes IPv4 Identification misassociations even worse
- IPv6 fragmentation is an IPv6 extension header, and some paths unconditionally drop packets with IPv6 extension headers
- For both IPv4 and IPv6, minimum reassembly buffer sizes are too small to support reassembly of an encapsulated packet that contains a 1500 byte payload
Fragmentation Alternative 2: Fragment Encapsulated Packet

- Break encapsulated packet into N pieces
- Append an identical IP header at the beginning of each piece
- Encapsulate each fragment in an Encapsulation IP header plus shim

<table>
<thead>
<tr>
<th>Encapsulation IP Header</th>
<th>Encapsulation IP Header</th>
<th>Encapsulation IP Header</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encapsulation Shim</td>
<td>Encapsulation Shim</td>
<td>Encapsulation Shim</td>
</tr>
<tr>
<td>Encapsulated IP Header</td>
<td>Encapsulated IP Header</td>
<td>Encapsulated IP Header</td>
</tr>
<tr>
<td>(MF=1)</td>
<td>(MF=1)</td>
<td>(MF=0)</td>
</tr>
<tr>
<td>Encapsulated Packet Body</td>
<td>Encapsulated Packet Body</td>
<td>Encapsulated Packet Body</td>
</tr>
<tr>
<td>First Piece</td>
<td>Second Piece</td>
<td>Final Piece</td>
</tr>
</tbody>
</table>
Alternative 2 Issues

- Only available for encapsulated IPv4 packets, since IPv6 does not allow in-the-network fragmentation
**Fragmentation Alternative 3: Tunnel Fragmentation**

- Break encapsulated packet into N pieces
- Encapsulate each fragment in an Encapsulation IP header plus shim
- **Shim header has Identification, MF values**

<table>
<thead>
<tr>
<th>Encapsulation IP Header</th>
<th>Encapsulation IP Header</th>
<th>Encapsulation IP Header</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encapsulation Shim (MF=1)</td>
<td>Encapsulation Shim (MF=1)</td>
<td>Encapsulation Shim (MF=0)</td>
</tr>
<tr>
<td>Encapsulated IP Header</td>
<td>Encapsulated IP Header</td>
<td>Encapsulated IP Header</td>
</tr>
<tr>
<td>Encapsulated Packet Body</td>
<td>Encapsulated Packet Body</td>
<td>Encapsulated Packet Body</td>
</tr>
<tr>
<td>First Piece</td>
<td>Second Piece</td>
<td>Final Piece</td>
</tr>
</tbody>
</table>
Alternative 3 Characteristics

• Avoids all issues identified for Alternatives 1 and 2
• Allows specification of a minimum reassembly buffer large enough to reassemble a 1500 byte encapsulated packet
Tunnel Fragmentation References

• First proposed in RFC2764
• Proposed Tunnel Fragmentation Extension for GRE:
• Proposed Tunnel Fragmentation Extension for GUE: