draft-perez-radext-radius-fragmentation
Objective

• Define a mechanism to deal with RADIUS packets over 4 KB, that are possible when:
  – Many different attributes are present
  – Big extended fragmented attributes are present (e.g. SAML-Message)
  – A combination of both

• Avoid using an out-of-band mechanism to retrieve data
  – Use a single trust infrastructure → RADIUS

• Compatible with existing specifications for intra-packet fragmentation
  – i.e. ietf-radext-radius-extensions
Overview

- A RADIUS peer willing to send a > 4KB RADIUS packet will do the following process:

  1) Divide the packet into smaller packets (called chunks) < 4 KB
     - If the last attribute of a chunk has flag “M” set (ietf-radext-radius-extensions) it is marked with an additional flag “T” to indicate this is not an error

  2) A new attribute called More-Data-Pending is included in every chunk, except on the last one
     - Indicates that more data is required to rebuild the original packet
     - Equivalent to the flag M, but at a packet-level

  3) Send chunks to the receiver in order
     - Using Access-Request/Access-Challenge exchanges
     - Receiver do not process chunks until the original packet is completely rebuilt
**More-Data-Pending**

- **Format:**

<table>
<thead>
<tr>
<th>Type</th>
<th>Length</th>
<th>Extended-Type</th>
<th>Value</th>
</tr>
</thead>
</table>

  - Value: Not defined yet
Example: Access-Request

- Maximum packet length = 8 attributes

**Access-Request** = User-Name, Calling-Station-Id, Data1[M], Data2[M], Data3[M], Data4[M], Data5[M], Data6[M], Data7[M], Data8[M], Data9[M], Data10, Other1[M], Other2[M], Other3

```
+--------+       +--------+
|  NAS   |       |  AS    |
+--------+       +--------+
```

```
| Access-Request(User-Name, Calling-Station-Id, |
| Data1[M], Data2[M], Data3[M], Data4[M], |
| Data5[MT], More-Data-Pending) |
```

```
+--------------------------+---------->
| Access-Challenge(State1) |
|<--------------------------|

```
```
| Access-Request(State1, Data6[M], Data7[M], Data8[M], |
| Data9[M], Data10, Other1[MT], |
| More-Data-Pending |
```

```
+--------------------------+---------->
| Access-Challenge(State2) |
|<--------------------------|

```
```
| Access-Request(State2, Other2[M], Other3) |
```

Example: Access-Challenge

- Maximum packet length = 8 attributes

\[
\text{Access-Challenge} = \text{Data1}\[M\], \text{Data2}\[M\], \text{Data3}\[M\], \text{Data4}\[M\], \text{Data5}\[M\], \text{Data6}\[M\], \text{Data7}\[M\], \text{Data8}\[M\], \text{Data9}\[M\], \text{Data10}, \text{Other1}\[M\], \text{Other2}\[M\], \text{Other3}
\]

```
+-----+      +-----+
| NAS |      | AS  |
+-----+      +-----+
        |
        |
        |
        |
        |
        |
        |
+-----+      +-----+
|      |      |
|      |      |
|      |      |
|      |      |
|      |      |
+-----+      +-----+
        |
        |
        |
        |
        |
        |
```

Access-Challenge(Data1[M],Data2[M],Data3[M], Data4[M],Data5[M],Data6[M], Data7[M], Data8[M], Data9[M], Data10, Other1[M], Other2[M], Other3)

Access-Request(State1)

Access-Challenge(Data7[M],Data8[M],Data9[M], Data10,Other1[M],Other2[M], Other2[MT], More-Data-Pending,State2)

Access-Request(State2)

Access-Challenge(Other3)
Example: Access-Accept

- Some attributes are allowed to appear ONLY in Access-Accept packets
  - They cannot be present in Access-Challenge packets
  - Previous solution not directly applicable

- Solution:
  - AS sends Access-Accept including all these attributes
    - Service-Type="Authorize-Only"
    - Include State attribute
  - NAS requests rest of information before granting access
    - Access-Request/Access-Challenge exchanges
    - Last chunk is Access-Accept with the actual Service-Type
Example: Access-Accept

- Maximum packet length = 8 attributes

\[
\text{Access-Accept} = \text{User-Name, Service-Type}[X], \text{Framed-IP-Address, Data1[M], Data2[M], Data3[M], Data4[M], Data5[M], Data6[M], Data7[M], Data8[M], Data9[M], Data10}
\]

```
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>NAS</td>
<td>AS</td>
</tr>
</tbody>
</table>
```

```
Access-Accept(User-Name, Service-Type[AuthOnly], Framed-IP-Address, State1)
```

```
Access-Request(State1)
```

```
Access-Challenge(Data1[M], Data2[M], Data3[M], Data4[M], Data5[M], Data6[MT], More-Data-Pending, State2)
```

```
Access-Request(State2)
```

```
Access-Accept(Data7[M], Data8[M], Data9[M], Data10, Service-Type[X])
```

```
Proxies

• Can they introduce attributes (e.g. Proxy-State)?
  – As long as they do not exceed the 4KB limit for each chunk
  – Sender may need to leave enough room for extra attributes
    • E.g. Make chunks < 3 KB instead of 4 KB
    • A new attribute similar to Framed-MTU might be used for this purpose

• Can they modify attributes?
  – Proxy interacts with sender to obtain the original packet
    • Need to hold state until all chunks are received
  – Proxy modifies attributes and generates new packet
  – Proxy delivers new chunks to the receiver
Proxies

- Can proxies modify attributes?

```
  +-----+  +-----+  +-----+
  | NAS |  | Proxy |  | AS  |
  +-----+  +-----+  +-----+

  Access-Request(User-Name, Calling-Station-Id, Data1[M], Data2[M], Data3[M], Data4[M], Data5[MT], More-Data-Pending)

  Access-Challenge(State1)

  Access-Request(State1, Data6[M], Data7[M], Data8[M], Data9[M], Data10)

  Access-Request(User-Name, Calling-Station-Id, Data1[M], Data2[M], Data3[M], Data4[M], Data5[MT], More-Data-Pending)

  Access-Challenge(State2)

  Access-Request(State2, Data6[M], Data7)
```
Does the WG consider this proposal interesting to become a WG item?